D209: Data Mining I

Task 1

Lucas Benish

Western Governors University

# Part 1: Research Question

# Describe the purpose of this data mining report by doing the following:

## Propose **one** question relevant to a real-world organizational situation that you will answer using **one** of the following classification methods:

## *k*-nearest neighbor (KNN)

## Naïve Bayes

The question that I am interested in answering is as follows:

Using our current dataset, can we predict if a patient will readmit to the hospital based on their medical history/comorbidities and/or other patient factors?

This will be investigated using k-nearest neighbor (KNN) classification. This question qualifies as a classification task due to the outcome of interest being either Yes (patient did ReAdmit) or No (patient did not ReAdmit).

## Define **one** goal of the data analysis. Ensure that your goal is reasonable within the scope of the scenario and is represented in the available data.

My main goal of this data analysis is to create a classification model, using KNN, that will predict readmission based upon patient medical history/comorbidities, and is not overfit to the original dataset. This latter goal will ideally be accomplished via splitting the data in training and test groups, as well as using cross-validation. This will ensure that the model is tested on “new” data that was split off from the original dataset and was not used to train the model.

# Part II: Method Justification

# Explain the reasons for your chosen classification method from part A1 by doing the following:

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

The method that I’ve chosen, k-Nearest Neighbors (KNN) classifies a data point based on a pre-specified number (k) of surrounding points. The method classifies a data point as the value consistent with the majority of its closest neighbors. For example, with my current dataset, if I had set k=3 then a data point would be classified as ReAdmis[Yes] if 2 out of 3 of its closest neighbors were also ReAdmis[Yes].

## Summarize **one** assumption of the chosen classification method

One assumption of k-Nearest Neighbor is that similar data points are found near each other (Harrison, 2018). It is thought that even though this will not be true for all data points, it will be generally true for a whole data set.

## List the packages or libraries you have chosen for Python or R, and justify how *each* item on the list supports the analysis

The libraries that I have chosen to use within R are as follows:

readr: used for importing dataset from csv

class: used for performing KNN classification task

dplyr: utilized for easier syntax, piping functions/code

ggplot2: used for visualizing relationships between explanatory variables

visdat: used for visualizing missingness during exploration

naniar: used for cleaning missingness found during exploration

fastDummies: used to create dummy variable columns from my categorical variables

caret: used to perform classification model with use of cross-validation

pROC: used to analyze and display ROC/determine AUC

# Part III: Data Preparation

# Perform data preparation for the chosen data set by doing the following:

## Describe **one** data preprocessing goal relevant to the classification method from part A1.

One goal of data preprocessing for KNN would be normalizing the data. This will deter values that typically would be on a larger scale from having a larger influence on predictions. An example of this would be a patient’s income vs a patient’s initial stay in the hospital. In this dataset, income can sometimes be upwards of ~200,000 dollars, while stay in the hospital is upwards of ~70 days. To stop any specific numeric variable, in this case income, from having a larger influence on the model, all numeric values will be normalized so that they are on a 0 to 1 scale. This is performed utilizing a function that will be written manually.

## Identify the initial data set variables that you will use to perform the analysis for the classification question from part A1, and classify *each* variable as continuous or categorical.

The explanatory variables that I will be using in the classification model have been selected based on their use in the previous logistic regression for D208 Task 2. These are variables used within the reduced logistic regression model. Their use in the reduced model was chosen due to their significance within the initial logistic regression model that included all potential explanatory variables. The variables I have selected are as follows, with their initial categorical or continuous designation:

Response variable:

ReAdmis: categorical

Explanatory variables:

Children: continuous

Age: continuous

Initial\_admin: categorical

HighBlood: categorical

Stroke: categorical

Complication\_risk: categorical

Arthritis: categorical

Diabetes: categorical

Anxiety: categorical

Reflux\_esophagitis: categorical

Asthma: categorical

Services: categorical

Initial\_days: continuous

It should also be noted that as part of preprocessing for this data, all the above categorical variables will be made into numeric variables, and from there dummy variables will be created.

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

Much of the data cleaning and preparation process for this task is similar to D208 – Task 2 on logistic regression. This is also a classification task and this one builds upon the preparation and logistic regression done in D208. For this reason, visualization comparisons were performed later in my code after extra variables were already removed. In D208, visualization was performed earlier in the code for comparison of all explanatory variables with the response variable.

The first step of this task was simply uploading the data into my R studio environment. This was accomplished using the readr package and the following code:

#import dataset

medical\_clean <- read\_csv("C:/Users/lgben/OneDrive/Desktop/MSDA/D209 - Data Mining I/Task 1 - Classification Analysis/medical\_clean.csv")

View(medical\_clean)

The next steps that I took in preparing my data for this classification were viewing the overall structure of my data, visualizing the missingness, determining presence of duplicates, and determining presence of outliers. Prior to discovering outliers, I created a copy data frame in case something went awry with outlier removal. It was determined that there were no missing values within this dataset and no duplicates. This was accomplished by the following code:

#assessing missingness and duplicates

vis\_miss(medical\_clean)

str(medical\_clean)

sum(duplicated(medical\_clean))

#general glimpse of dataset

summary(medical\_clean)

#copy data set for outliers if needed

medical\_clean2 <- medical\_clean

Outlier presence was determined based on z-scores, with any z-score greater than 3 or less than -3 standard deviations from the mean being classified as an outlier. It was determined that there were 467 unique observations with outliers. Outliers were removed by creating a new dataframe.

Similarly to the prior course, I decided to remove all outliers from the dataset. The reason for this is that the total number of outliers, 467, is less than 5% of the dataset. This would leave me with over 95% of the data remaining and reduces the need to determine if some outliers were worth keeping. If I determined independently which outliers would be worth keeping, I would be introducing my own personal bias to the dataset. Outlier detection and removal was accomplished by the following code:

#z-score columns

medical\_clean2$children\_z <- scale(x=medical\_clean2$Children)

medical\_clean2$age\_z <- scale(x=medical\_clean2$Age)

medical\_clean2$income\_z <- scale(x=medical\_clean2$Income)

medical\_clean2$vitd\_levels\_z <- scale(x=medical\_clean2$VitD\_levels)

medical\_clean2$doc\_visits\_z <- scale(x=medical\_clean2$Doc\_visits)

medical\_clean2$full\_meals\_eaten\_z <- scale(x=medical\_clean2$Full\_meals\_eaten)

medical\_clean2$vitd\_supp\_z <- scale(x=medical\_clean2$vitD\_supp)

medical\_clean2$initial\_days\_z <- scale(x=medical\_clean2$Initial\_days)

medical\_clean2$totalcharge\_z <- scale(x=medical\_clean2$TotalCharge)

medical\_clean2$additional\_charges\_z <- scale(x=medical\_clean2$Additional\_charges)

#outlier vectors

children\_outliers <- which(medical\_clean2$children\_z >3 | medical\_clean2$children\_z < -3)

age\_outliers <- which(medical\_clean2$age\_z >3 | medical\_clean2$age\_z < -3)

income\_outliers <- which(medical\_clean2$income\_z >3 | medical\_clean2$income\_z < -3)

vitd\_levels\_outliers <- which(medical\_clean2$vitd\_levels\_z >3 | medical\_clean2$vitd\_levels\_z < -3)

doc\_visits\_outliers <- which(medical\_clean2$doc\_visits\_z >3 | medical\_clean2$doc\_visits\_z < -3)

full\_meals\_eaten\_outliers <- which(medical\_clean2$full\_meals\_eaten\_z >3 | medical\_clean2$full\_meals\_eaten\_z < -3)

vitd\_supp\_outliers <- which(medical\_clean2$vitd\_supp\_z >3 | medical\_clean2$vitd\_supp\_z < -3)

initial\_days\_outliers <- which(medical\_clean2$initial\_days\_z >3 | medical\_clean2$initial\_days\_z < -3)

total\_charge\_outliers <- which(medical\_clean2$totalcharge\_z >3 | medical\_clean2$totalcharge\_z < -3)

additional\_charges\_outliers <- which(medical\_clean2$additional\_charges\_z >3 | medical\_clean2$additional\_charges\_z < -3)

#treating outliers

unique\_outliers <- unique(c(children\_outliers, doc\_visits\_outliers, full\_meals\_eaten\_outliers, income\_outliers, vitd\_levels\_outliers, vitd\_supp\_outliers))

medical\_clean3 <- medical\_clean2[-unique\_outliers, ]

medical\_clean3 <- subset(medical\_clean3, select = -c(age\_z, income\_z, vitd\_levels\_z, doc\_visits\_z, full\_meals\_eaten\_z, vitd\_supp\_z, initial\_days\_z, totalcharge\_z, additional\_charges\_z, children\_z))

#removing outlier vectors/objects from environment

remove(additional\_charges\_outliers, age\_outliers, children\_outliers, doc\_visits\_outliers, full\_meals\_eaten\_outliers, income\_outliers, initial\_days\_outliers, total\_charge\_outliers, unique\_outliers, vitd\_levels\_outliers, vitd\_supp\_outliers)

After outliers were removed from the dataset and programming environment, I created a numeric response variable and reduced the overall dataset to only the previously noted significant explanatory variables using the following code. This was done to create visualizations, as well as for creation of a classification model.

#creating numeric response variable

medical\_clean3$ReAdmisNumeric <- as.numeric(as.factor(medical\_clean3$ReAdmis))

medical\_clean3$ReAdmisNumeric <- medical\_clean3$ReAdmisNumeric-1

hist(medical\_clean3$ReAdmisNumeric)

#reducing dataset to only previously noted significant explanatory variables

medical\_clean4 <- subset(medical\_clean3, select = c(Children, Age, Initial\_admin, HighBlood, Stroke, Complication\_risk, Arthritis, Diabetes, Anxiety, Reflux\_esophagitis, Asthma, Services, Initial\_days, ReAdmisNumeric, ReAdmis))

In the previous course I visualized all potential explanatory variables independently, in univariate visualizations, and with the response variable in bivariate visualizations, prior to removal of outliers. For the purposes of this course, I only visualized those explanatory variables that were previously determined to be significant as a reduced logistic regression model had already been created in D208. Therefore after all extraneous explanatory variables were removed, I then visualized my remaining explanatory variables in univariate visualizations and bivariate visualizations to refamiliarize myself with them. This was accomplished using the following code:

#visualizing only those pertinent variables from prior logreg

#univariate viz

ggplot(medical\_clean4, aes(x=ReAdmisNumeric)) + geom\_histogram(bins=3)

ggplot(medical\_clean4, aes(x=Children)) + geom\_histogram(binwidth=1)

ggplot(medical\_clean4, aes(x=Age)) + geom\_histogram(binwidth = 2)

ggplot(medical\_clean4, aes(x=Initial\_admin)) + geom\_bar()

ggplot(medical\_clean4, aes(x=HighBlood)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Stroke)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Complication\_risk)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Arthritis)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Diabetes)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Anxiety)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Reflux\_esophagitis)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Asthma)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Services)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Initial\_days)) + geom\_histogram()

#bivariate viz

ggplot(medical\_clean4, aes(x=Children, y=ReAdmisNumeric)) + geom\_point() + geom\_smooth(method='glm', se=FALSE)

ggplot(medical\_clean4, aes(x=Age, y=ReAdmisNumeric)) + geom\_point() + geom\_smooth(method='glm', se=FALSE)

ggplot(medical\_clean4, aes(x=Initial\_admin, fill=ReAdmis)) + geom\_bar()

ggplot(medical\_clean4, aes(x=HighBlood, fill=ReAdmis)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Stroke, fill=ReAdmis)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Complication\_risk, fill=ReAdmis)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Arthritis, fill=ReAdmis)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Diabetes, fill=ReAdmis)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Anxiety, fill=ReAdmis)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Reflux\_esophagitis, fill=ReAdmis)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Asthma, fill=ReAdmis)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Services, fill=ReAdmis)) + geom\_bar()

ggplot(medical\_clean4, aes(x=Initial\_days, y=ReAdmisNumeric)) + geom\_point() + geom\_smooth(method='glm', se=FALSE)

Following this visualization, I then created dummy variable columns. The reason for this is that KNN performs best with variables that are on consistent interval scales, or where all intervals are the same length (Marsja, 2020). Dummy coding is also useful for variables that may not have an inherent rank, such as my variable Initial\_admin, where a patient may have been admitted from the ER, from observation, or an elective admission. There is no inherent rank between these, so this variable would benefit from dummy coding.

While creating the dummy columns, I removed the initial categorical columns. After this, I also removed the initial ReAdmis column while keeping ReAdmisNumeric that I had just created. This was done as follows:

#creating dummy explanatory variable columns

medical\_clean5 <- dummy\_cols(medical\_clean4, select\_column = c('Initial\_admin', 'HighBlood', 'Stroke', 'Complication\_risk', 'Arthritis', 'Diabetes', 'Anxiety', 'Reflux\_esophagitis', 'Asthma', 'Services'), remove\_selected\_columns = TRUE)

medical\_clean5 <- subset(medical\_clean5, select = -c(ReAdmis))

As the final step of data preprocessing/preparation, I normalized data that was numeric. The columns this applied to were Age, Children, and Initial\_days. This makes it so that each variable is now on a 0-1 scale and removes the possibility of variables with larger numeric values providing more influence on the model. After normalized columns were created, the initial columns were removed. This was done as follows:

#normalizing numeric variables that need it (not dummy coded)

normalize <- function(x) {

return((x - min(x)) / (max(x) - min(x)))

}

medical\_clean5$children\_normal <- normalize(medical\_clean5$Children)

medical\_clean5$age\_normal <- normalize(medical\_clean5$Age)

medical\_clean5$initial\_days\_normal <- normalize(medical\_clean5$Initial\_days)

medical\_clean5 <- subset(medical\_clean5, select = -c(Children, Age, Initial\_days))

## Provide a copy of the cleaned data set.

A copy of the cleaned data set, medical\_clean5, will be attached to my submission. This was written to my folder using the following code:

#writing cleaned/pre-processed set to folder

write.csv(medical\_clean5, 'C:\\Users\\lgben\\OneDrive\\Desktop\\MSDA\\D209 - Data Mining I\\Task 1 - Classification Analysis\\medical\_clean5.csv')

# Part IV: Analysis

# Perform the data analysis and report on the results by doing the following:

## Split the data into training and test data sets and provide the file(s)

The cleaned dataset, medical\_clean5, was split into a training set, medical\_clean5\_train, and testing set, medical\_clean5\_test. R Documentation was consulted for the sample() function (DataCamp, n.d.) These will be attached to submission and were attained via the code below:

#set seed for future needs

set.seed(42)

#train-test-split

sample\_rows <- sample(nrow(medical\_clean5), nrow(medical\_clean5)\*0.8)

n\_80\_20\_split <- round(nrow(medical\_clean5) \* 0.8, 0)

medical\_clean5\_train <- medical\_clean5[sample\_rows, ]

medical\_clean5\_test <- medical\_clean5[-sample\_rows, ]

#writing split data sets to folder

write.csv(medical\_clean5\_train, 'C:\\Users\\lgben\\OneDrive\\Desktop\\MSDA\\D209 - Data Mining I\\Task 1 - Classification Analysis\\medical\_clean5\_train.csv')

write.csv(medical\_clean5\_test, 'C:\\Users\\lgben\\OneDrive\\Desktop\\MSDA\\D209 - Data Mining I\\Task 1 - Classification Analysis\\medical\_clean5\_test.csv')

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

The analysis technique that I used to model my data was k-Nearest Neighbors (KNN) to predict readmission status. I performed this analysis two separate ways, using two different R packages.

Initially I used the class package and the knn() function, and I performed three different knn() calls, each with different k values of 3, 5, and 7. R Documentation was once again consulted for this (DataCamp, n.d.). Prior to doing this, I had to further split my already created train and test datasets to remove the actual ReAdmis values from the dataset and save this column as a vector. Following this, I created test2 and train2 datasets that included all the model data except for the actual ReAdmis values. This can be seen with the code below, with the average accuracy noted via mean() calculation of predictions matching actual values.

Next, I utilized the caret package and train() followed by predict() functions. Prior to training my model, I needed to make my response variable once again into a factor, and did this by creating another training dataset, medical\_clean5\_train3, and another test dataset, medical\_clean5\_train3. Then, I utilized trainControl() to set my cross validation to 5 folds. Pre-processing was not included as part of my train() call, as this was already completed during my data exploration and cleaning phase. Following fitting this model, predictions were made using predict() on my knn\_caret object and my medical\_clean5\_test3 dataset. The code, and accuracy, can be seen below:

Each model performed essentially the same, with the accuracy on the testing data for all of them being ~97%, regardless of package used. This was determined by taking the average of test\_actual\_readmis\_numeric == k\_i, where i is the number of nearest neighbors. Screenshots are included below, as I would consider this one of the intermediate calculations performed.

A picture containing text, indoor, monitor, screenshot

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

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

The code that I utilized for classification analysis described above in D2 is as follows:

#creating response vectors, new dataframes without responses

train\_actual\_readmis\_numeric <- medical\_clean5\_train$ReAdmisNumeric

medical\_clean5\_train2 <- subset(medical\_clean5\_train, select = -c(ReAdmisNumeric))

test\_actual\_readmis\_numeric <- medical\_clean5\_test$ReAdmisNumeric

medical\_clean5\_test2 <- subset(medical\_clean5\_test, select = -c(ReAdmisNumeric))

#creating different knn models

k\_3 <- knn(train=medical\_clean5\_train2, test=medical\_clean5\_test2, cl=train\_actual\_readmis\_numeric, k=3)

mean(test\_actual\_readmis\_numeric == k\_3)

#0.9737

k\_5 <- knn(train=medical\_clean5\_train2, test=medical\_clean5\_test2, cl=train\_actual\_readmis\_numeric, k=5)

mean(test\_actual\_readmis\_numeric == k\_5)

#0.9775

k\_7 <- knn(train=medical\_clean5\_train2, test=medical\_clean5\_test2, cl=train\_actual\_readmis\_numeric, k=7)

mean(test\_actual\_readmis\_numeric == k\_7)

#0.9775

#see no difference in average accuracy of k\_5 and k\_7

#creating new dataframes with ReAdmis as factor

medical\_clean5\_train3 <- medical\_clean5\_train

medical\_clean5\_train3$ReAdmisFactor <- factor(medical\_clean5\_train3$ReAdmisNumeric)

medical\_clean5\_train3 <- subset(medical\_clean5\_train3, select = -c(ReAdmisNumeric))

medical\_clean5\_test3 <- medical\_clean5\_test

medical\_clean5\_test3$ReAdmisFactor <- factor(medical\_clean5\_test3$ReAdmisNumeric)

medical\_clean5\_test3 <- subset(medical\_clean5\_test3, select = -c(ReAdmisNumeric))

#using caret instead of class to allow for cross-val

trcontrol <- trainControl(method = 'cv', number = 5)

knn\_caret <- train(

ReAdmisFactor ~.,

data = medical\_clean5\_train3,

method = 'knn',

trControl=trcontrol,

)

knn\_caret\_test\_readmis\_predictions <- predict(knn\_caret, medical\_clean5\_test3)

mean(knn\_caret\_test\_readmis\_predictions == test\_actual\_readmis\_numeric)

#0.9780

# Part V: Data Summary and Implications

# Summarize your data analysis by doing the following:

## Explain the accuracy and the area under the curve (AUC) of your classification model.

The accuracy of my model can be determined by utilizing a confusion matrix. A confusion matrix is comprised of the correctly and incorrectly predicted ReAdmis status of the individuals within my data. Accuracy is calculated from the formula (true\_positive + true\_negative)/(true\_positive + true\_negative + false\_negative + false\_positive).

The receiver operator characteristic (ROC) curve is a measure of a model’s sensitivity on the y-axis relative to 1 – specificity (false positive rate) on the x-axis. The area under the curve (AUC) is a measure of the model’s accuracy and shows the relationship between true positives (sensitivity) and false positives. It can be used as a measure of accuracy of binary classification tasks (Sanghun Nahm, 2022)

The calculated accuracy of my model is 0.9780, using the formula from above to manually calculate it. The AUC, as determined via auc() on an ROC object, was 0.996. Both accuracy and AUC demonstrate that the KNN model created is very accurate.

The code that I utilized for everything discussed within this section is as follows:

#accuracy

conf\_mat <- table(medical\_clean5\_test3$ReAdmisFactor, knn\_caret\_test\_readmis\_predictions)

true\_neg <- conf\_mat[1, 1]

true\_pos <- conf\_mat[2, 2]

false\_neg <- conf\_mat[2, 1]

false\_pos <- conf\_mat[1, 2]

accuracy <- (true\_neg + true\_pos)/(true\_neg + true\_pos + false\_neg + false\_pos)

#creating ROC curves

knn\_caret\_prob\_pred <- predict(knn\_caret, medical\_clean5\_test3, type='prob')

ROC <- roc(medical\_clean5\_test3$ReAdmisFactor, knn\_caret\_prob\_predictions$`1`)

plot(ROC)

AUC <- auc(ROC)

## Discuss the results and implications of your classification analysis

The results of this classification model are very accurate, as noted by both the accuracy and area under the curve being greater than 0.99. This likely means that either the data in my training and testing data sets are too similar or the model was overfit on our training data.

For the time being, this model could potentially be used to predict whether a new patient that enters the hospital may readmit at a future date. This decision is based upon the presence or absence of the factors included within the model (complication risk, admission status, services, etc.) for that specific patient.

## Discuss **one** limitation of your data analysis

One limitation of this classification model is that I did not utilize all explanatory variables. I did this to limit the power needed by my computer to perform the analysis and utilize only previously noted significant variables. However, because I was using a different classification model, there may have been some interactions missed from the explanatory variables that were not utilized.

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

I believe that the best course of action would be to utilize the model for making predictions and assisting with making business decisions. By using the model to predict which patients are more likely to readmit to the hospital in the future we can provide targeted, specific education and home management tasks to those patients to reduce readmission risk.

Further, continued data gathering should be performed. This model should not be considered “complete” as patients are going to continue to come to the hospital and new risk factors may be more important in the future. For that reason, I think re-evaluating the model in the future, as well as possibly creating a new model and comparing its accuracy to the old one, would be a good next step in 6-12 months.

# Part VI: Demonstration

# Provide a Panopto video recording that includes a demonstration of the functionality of the code used for the analysis and a summary of the programming environment.

## *Note: The audiovisual recording should feature you visibly presenting the material (i.e., not in voiceover or embedded video) and should simultaneously capture both you and your multimedia presentation*

Please see Panopto video.

# Record the web sources used to acquire data or segments of third-party code to support the analysis. Ensure the web sources are reliable.

# Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

1. DataCamp. (n.d.). *Knn: K-nearest neighbour classification*. RDocumentation. Retrieved November 11, 2022, from <https://www.rdocumentation.org/packages/class/versions/7.3-20/topics/knn>
2. DataCamp. (n.d.). *Sample: Random samples and permutations*. RDocumentation. Retrieved November 11, 2022, from <https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/sample>
3. Harrison, O. (2018, September 10). *Machine learning basics with the K-nearest neighbors algorithm*. Towards Data Science. Retrieved November 7, 2022, from <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>
4. Marsja, E. (2020, May 24). *How to create dummy variables in R (with examples)*. Erik Marsja. Retrieved November 7, 2022, from <https://www.marsja.se/create-dummy-variables-in-r/>
5. Sahngun Nahm, F. (2022, January 18). *Receiver operating characteristic curve: Overview and practical use for clinicians*. Korean journal of anesthesiology. Retrieved November 7, 2022, from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8831439/>

# Demonstrate professional communication in the content and presentation of your submission.