**Performance Assessment: D209- Classification Analysis**

**A. Research Question**

**1.** For this assessment task, my research question is as follows: can we predict readmissions based on whether or not the patient has a condition (i.e., High blood pressure, Diabetes, Anxiety, etc.). I will answer this question using the Naive Bayes classification method.

**2.** The goal of this analysis is to determine whether or not we can predict whether or not a patient is more or less likely to be readmitted to the hospital within a month of discharge if they also suffer from one or more of the following condition or variables: Soft\_drink, HighBlood, Stroke, Overweight, Arthritis, Diabetes, Hyperlipidemia, BackPain, Anxiety, Allergic\_rhinitis, Reflux\_esophagitis, and Asthma.

**B. Method Justification**

**1.** The Naïve Bayes algorithm is used to predict whether the variables in the data set can predict whether or not the target variable is affected. The algorithm works by assuming the variables are independent of each other (Ray 2023) and trying to determine the probability that a patient is readmitted to the hospital based on whether or not that patient has answered “Yes” to one of the condition variables. In other words, what is the probability that you are readmitted given that you have X condition. In this scenario, the outcomes that I expect are that the probability of readmittance is higher if the patient has one of the condition variables.

**2.** One assumption of Naive Bayes is that “all features in the input data are independent of each other” (Ray 2023). This is stating that all of the variables that we are testing are not related to or affect one another.

**3.** The following list is of the packages that I will be using in R and their respective uses for the analysis:

* Caret: for the naïve bayes algorithm
* klaR: for visualizations for the naïve bayes algorithm
* Gains: for creating our gains table
* pROC: for determining the area under the curve of the Naive Bayes model

**C. Data Preparation**

**1.** In order to clean the data and prepare it for analysis, I will first delete all columns in the CSV file that will not be used in the model, which will be every variable that is not mentioned below in question **2**. This step will be done in Microsoft Excel and not R as it is much faster and more convenient to delete the unnecessary columns directly instead of using code to achieve this step. Then I will copy and paste the columns, so they line up next to each other in order instead of having blank spaces where numerical data or irrelevant data used to be. After this, I will check for null values and replace them if necessary. Since all my variables are categorical variables, any null values will be replaced with a “No” since you cannot assume a patient has any particular condition variable. Finally, I will revalue the ReAdmis variable. Since readmissions is my target variable, in order to properly perform the Naive Bayes algorithm and model, I will change all “No” responses to a 0 and all “Yes” responses to a 1.

**2.** The following is a list of the variables that I will be using and their respective classifications:

* ReAdmis (target variable): categorical
* Soft\_drink: categorical
* HighBlood: categorical
* Stroke: categorical
* Overweight: categorical
* Arthritis: categorical
* Diabetes: categorical
* Hyperlipidemia: categorical
* BackPain: categorical
* Anxiety: categorical
* Allergic\_rhinitis: categorical
* Reflux\_esophagitis: categorical
* Asthma: categorical

**3.** Firstly, I opened the given CSV file and simply deleted all of the columns that were not necessary for this analysis. Afterwards, I imported the CSV file into R. Using the colSums(is.na) function, I then determined that there were no variables with missing data. The last step was to change the ReAdmis variable into 0s and 1s to represent No and Yes; after which, I then turned the variable into a numeric data. This process is demonstrated from the following screenshot:

A computer screen shot of a computer code

Description automatically generated

**4.** A copy of the cleaned data sheet has been attached alongside this written assessment.

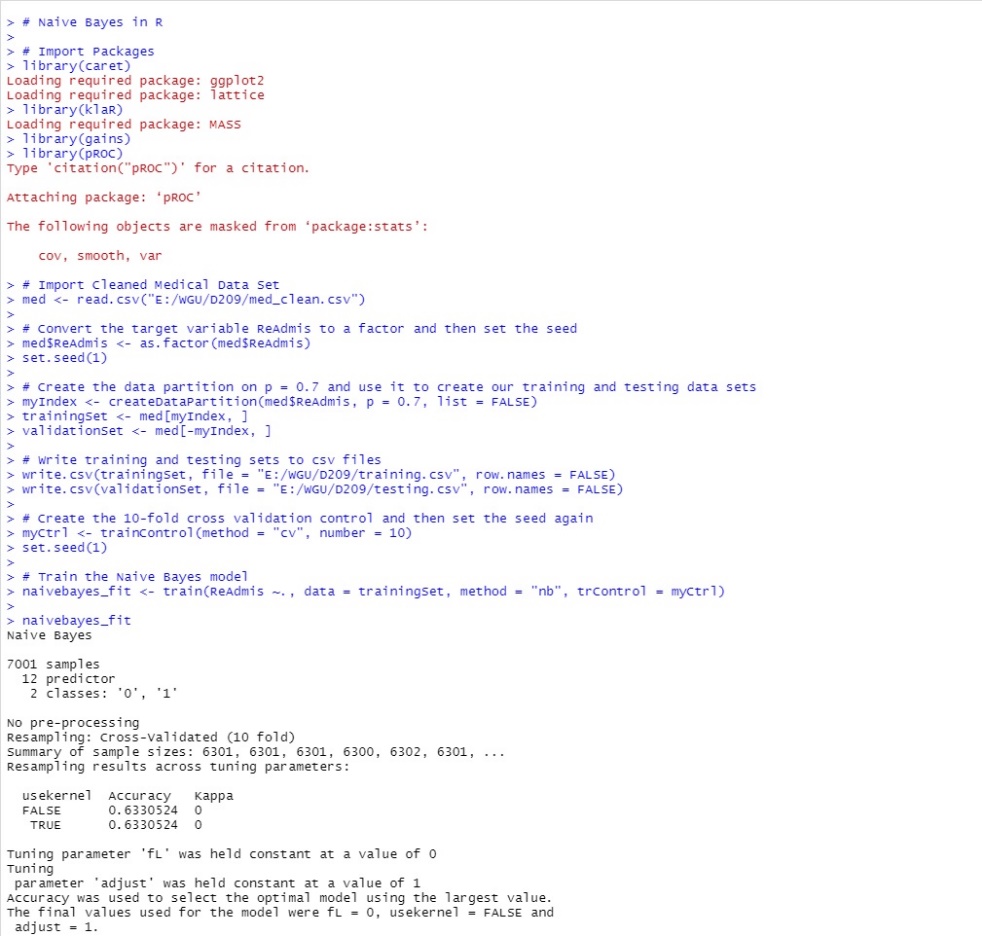
**D. Analysis**

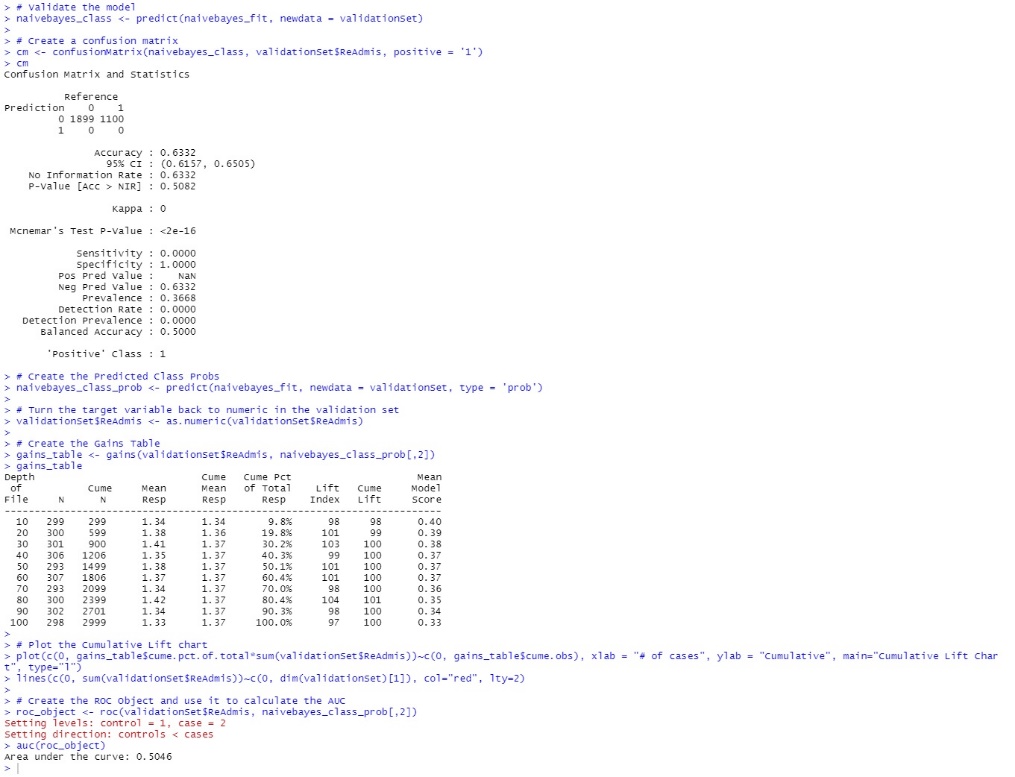
**1.** The training and testing data sets have been attached as CSV files alongside this written assessment.

**2.** Below is a list of the steps used to run the classification analysis (naïve bayes) algorithm in R:

* Import the packages from **B3**
* Import the cleaned data set from part **C**
* For this algorithm to work the target variable needs to be a factor so I turned the ReAdmis target variable into a factor and then set the seed, which makes the sets reproducible
* Created a data partition that splits the data set into a 70/30 ratio, and by using this partition, I split the data set into a training set and then a validation, or testing set, and then used the write.csv function to turn those into csv files to attach for part **D1**
* Created a cross folded validation control on a 10-fold scale. This divides the training set from the previous step into smaller sets where the algorithm then picks which set works the best
* Create the initial naïve bayes model fit which is using the training set and is controlled using the cross-fold validation, and when we look at that, as you can see from the screenshots in the next question, the fit is only 63% accurate on both the false and true kernels.
* Then we validate the training model using the validation set to predict the class from the training set and then we turn that into a confusion matrix that again shows a 63% accuracy.
* The next step is to create a gains table so first, I created a class of probabilities from the predicted class, then turn the ReAdmis variable back into numeric, and then create the gains table.
* Last step is to use the gains table to plot a cumulative lift chart, then the roc\_object which is used to calculate the AUC or area under the curve. This AUC can be seen on the cumulative lift chart.

**3.** Below are screenshots of all code and calculations performed in R as well as screenshots of the cumulative lift chart. No other intermediate calculations were used for this assessment. Everything was achieved through R.





(Straw, 2021).

Listed below is a copy of the code from R for the data preparation that can be executed data free:

> # Data Cleaning for D209 Part 1

>

> # Import Data

> med <- read.csv("E:/WGU/D209/Task 1/medical\_clean\_removal.csv")

>

> # Check for Missing Values

> colSums(is.na(med))

> # There are no missing values so we can proceed to the next step

>

> # Revalue the ReAdmis variable

> med$ReAdmis[med$ReAdmis == "Yes"] <- 1

> med$ReAdmis[med$ReAdmis == "No"] <- 0

>

> # Change the ReAdmis Variable to numeric

> med$ReAdmis <- as.numeric(med$ReAdmis)

>

> # Write into csv file

> write.csv(med, file = "E:/WGU/D209/Task 1/med\_clean.csv", row.names = FALSE)

Listed below is a copy of the code from R for the classification analysis that can be executed error-free:

> # Naive Bayes in R

>

> # Import Packages

> library(caret)

> library(klaR)

> library(gains)

> library(pROC)

> # Import Cleaned Medical Data Set

> med <- read.csv("E:/WGU/D209/Task 1/med\_clean.csv")

>

> # Convert the target variable into a factor and then set the seed

> med$ReAdmis <- as.factor(med$ReAdmis)

> set.seed(1)

>

> # Create the data partition on p = 0.7 and use it to create our training and testing data sets

> myIndex <- createDataPartition(med$ReAdmis, p = 0.7, list = FALSE)

> trainingSet <- med[myIndex, ]

> validationSet <- med[-myIndex, ]

>

> # Write training and testing sets to csv files

> write.csv(trainingSet, file = "E:/WGU/D209/training.csv", row.names = FALSE)

> write.csv(validationSet, file = "E:/WGU/D209/testing.csv", row.names = FALSE)

>

> # Create the 10-fold cross validation control and then set the seed again

> myCtrl <- trainControl(method = "cv", number = 10)

> set.seed(1)

>

> # Train the Naive Bayes model

> naivebayes\_fit <- train(ReAdmis ~., data = trainingSet, method = "nb", trControl = myCtrl)

> naivebayes\_fit

>

> # Validate the model

> naivebayes\_class <- predict(naivebayes\_fit, newdata = validationSet)

>

> # Create a confusion matrix

> cm <- confusionMatrix(naivebayes\_class, validationSet$ReAdmis, positive = '1')

> cm

> # Create the Predicted Class Probs

> naivebayes\_class\_prob <- predict(naivebayes\_fit, newdata = validationSet, type = 'prob')

>

> # Turn the target variable back to numeric in the validation set

> validationSet$ReAdmis <- as.numeric(validationSet$ReAdmis)

>

> # Create the Gains Table

> gains\_table <- gains(validationSet$ReAdmis, naivebayes\_class\_prob[,2])

> gains\_table

> # Plot the Cumulative Lift chart

> plot(c(0, gains\_table$cume.pct.of.total\*sum(validationSet$ReAdmis))~c(0, gains\_table$cume.obs), xlab = "# of cases", ylab = "Cumulative", main="Cumulative Lift Chart", type="l")

> lines(c(0, sum(validationSet$ReAdmis))~c(0, dim(validationSet)[1]), col="red", lty=2)

>

> # Create the ROC Object and use it to calculate the AUC

> roc\_object <- roc(validationSet$ReAdmis, naivebayes\_class\_prob[,2])

Setting levels: control = 1, case = 2

Setting direction: controls < cases

> auc(roc\_object)

**E. Data Summary and Implications**

**1.** The accuracy can be measured by observing the confusion matrix as pictured here:

A screenshot of a computer

Description automatically generated

By observing the confusion matrix, we can see the accuracy of the model is only 63%. This means that when we run the algorithm, it can only correctly predict the ReAdmis target variable approximately 63% of the time, which is low.

Next, we observe the area under the curve from the cumulative lift chart as pictured below:

A graph on a computer screen

Description automatically generated

By creating the ROC object, I calculated the AUC to be 0.5046 or approximately 50%. This calculation can be seen through the screenshot of the code from part **D3**. This is why it is almost impossible to detect the red line in the picture above, because they are almost identical lines. An AUC of 0.5046 signifies that there is no predictive ability within this model to accurately guess readmissions.

**2.** Since both the confusion matrix accuracy and the area under the curve in the cumulative lift chart are very low, there exists no implications between the target variable and the predictor variables. In other words, there is no statistical evidence that the various medical condition variables can be used to predict whether or not a patient is more likely to be readmitted into the hospital within a month of release.

**3.** One limitation of this data set is that the patients reason for admittance and readmittance are not included. A patient could have been admitted to the hospital for a reason totally unrelated to their condition, such as an injury, and readmitted later for the same reason.

**4.** Since the data provides no statistical evidence to predicting readmission, there is no course of action that can be recommended based on this algorithm. The only recommendation is to keep proceeding as normal and make no changes to how the patients who answer “Yes” to one or more of the condition variables are treated when they are admitted.

**F. Sources**

Ray, Sunil. “Naive Bayes Classifier Explained: Applications and Practice Problems of Naive Bayes Classifier.” Analytics Vidhya, 11 May 2023, [www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/#:~:text=The%20naive%20Bayes%20classifier%20assumes,true%20in%20real%2Dworld%20scenarios](http://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/#:~:text=The%20naive%20Bayes%20classifier%20assumes,true%20in%20real%2Dworld%20scenarios).

Straw, Eric. “Naive Bayes Model Building in R.” Vimeo, 5 March, 2021, vimeo.com/520220535.

No other sources were used in the creation of this assessment.