D209: Data Mining 1

Performance Assessment

Task 1

Logan Rosemeyer

Western Governor's University

D209: Data Mining 1

Dr. Eric Straw

January 15, 2023

Contents

A1: Proposal of Question

Is it possible to predict initial admission reason based on the conditions that the patient has using KNN, so the organization can reduce the number of emergency admissions by catching complications earlier?

A2: Defined Goal

The goal of the data is to see if we can accurately classify the patient's reason for initial admission based on the medical conditions they already have. If we can do that, we can make a plan to lower emergency admissions.

B1: Explanation of Prediction Method

Knn will look at all observations, and when trying to classify a new target, will look at the closest observations to the new target to predict what the new target will be classified as. The expected outcomes of my target variable, initial admission, are emergency admission, observation admission, and elective admission.

B2: Summary of Method Assumption

One assumption of knn is that similar things exist close together. This is pretty much the whole basis of knn. This algorithm uses distance between two data points to classify the variable you are trying to predict. (Grant, 7/21/19)

B3: Packages and Libraries List

The first package I uploaded was the dplyr package, which allowed me to use the select function to narrow down the columns I wanted to use. I also uploaded the ggplot2 package, which I used for visualizations of my selected columns. This gave me an idea of which diseases would have a larger effect on initial admission. The next package was the class package. This contains the knn function to run my model. Then I uploaded the caTools package to split my data into train and test. The next library I used was caret. This gave me the confusionMatrix function which I used to test model accuracy. The final library I used was pROC. This allowed me to find the AUC of a multi class variable.

C1: Data Preprocessing

One data preprocessing goal I had was to make my data all numeric. This will allow my model to use actual distance. I was unable to run the model when the variables were character types. Another goal I had was to create a data frame with only variables that I wanted for the model.

C2: Data Set Variables

All variables that I am using are categorical. They are initial admission, high blood pressure, stroke, overweight, arthritis, diabetes, hyperlipidemia, back pain, anxiety, allergic rhinitis, reflux esophagitis, and asthma. I did end up changing these to a numeric data with 1/0 instead of yes/no, but it still represents categorical data.

C3: Steps for Analysis

The first thing I did to prepare my data set was to load the csv into R, import the necessary libraries, and look at the head and structure of my data.

```
medical <- read.csv("D://MS DA//D209 Task 1//medical_clean.csv",
stringsAsFactors = T)

library(dplyr)

library(ggplot2)

library(class)

library(caTools)

library(caret)

head(medical)

str(medical)
```

The next step was to create a data frame with the variables I wanted to use to answer my question. I also changed yeses to 1s and noes to 0s in all of the disease variables.

```
disease <- medical %>% select(Initial_admin:Stroke, Overweight:Asthma)

disease[2:12] <- ifelse(disease[2:12] == "Yes", 1, 0)
```

I also ended up making initial admission numeric because I was running into problems running the model when it was a character data type.

```
# Elective Admission = 1

# Emergency Admission = 2

# Observation Admission = 3

disease[1] <- ifelse(disease[1] == "Elective Admission", 1,
ifelse(disease[1] == "Emergency Admission", 2, 3))
```

Next I checked for any duplicates or null values. If I had continuous variables, I would have also checked for outliers.

```
sum(duplicated(disease))

colSums(is.na(disease))
```

C4: Cleaned Data Set

Cleaned data set has been uploaded in the submission.

D1: Splitting the Data

The training and test data sets are uploaded in the submission. 80% of the data is training, 20% is test.

D2: Output and Intermediate Calculations

I used a confusion matrix to analyze my model. As you can see, there were only 3 incorrect predictions and the model was 99.9% accurate on the test data. This particular model used a k value of 4. I also tested models with k values of 2 and 3 to compare the accuracy of each. The other two confusion matrices are shown below as well. They had accuracies of 99.8%, which is still very good.

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3
         1  498    0    0
         2    3 1012    0
         3    0    0  487

Overall Statistics

               Accuracy : 0.9985
                 95% CI : (0.9956, 0.9997)
    No Information Rate : 0.506
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9976

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            0.9940   1.0000   1.0000
Specificity            1.0000   0.9970   1.0000
Pos Pred Value         1.0000   0.9970   1.0000
Neg Pred Value         0.9980   1.0000   1.0000
Prevalence             0.2505   0.5060   0.2435
Detection Rate         0.2490   0.5060   0.2435
Detection Prevalence   0.2490   0.5075   0.2435
Balanced Accuracy      0.9970   0.9985   1.0000
```

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3
         1  497    0    0
         2    4 1012    1
         3    0    0  486

Overall Statistics

               Accuracy : 0.9975
                 95% CI : (0.9942, 0.9992)
    No Information Rate : 0.506
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.996

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            0.9920   1.0000   0.9979
Specificity            1.0000   0.9949   1.0000
Pos Pred Value         1.0000   0.9951   1.0000
Neg Pred Value         0.9973   1.0000   0.9993
Prevalence             0.2505   0.5060   0.2435
Detection Rate         0.2485   0.5060   0.2430
Detection Prevalence   0.2485   0.5085   0.2430
Balanced Accuracy      0.9960   0.9975   0.9990
```


```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3
         1  497    0    0
         2    4 1012    1
         3    0    0  486

Overall Statistics

               Accuracy : 0.9975
                 95% CI : (0.9942, 0.9992)
    No Information Rate : 0.506
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.996

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            0.9920   1.0000   0.9979
Specificity            1.0000   0.9949   1.0000
Pos Pred Value         1.0000   0.9951   1.0000
Neg Pred Value         0.9973   1.0000   0.9993
Prevalence             0.2505   0.5060   0.2435
Detection Rate         0.2485   0.5060   0.2430
Detection Prevalence   0.2485   0.5085   0.2430
Balanced Accuracy      0.9960   0.9975   0.9990
```

D3: Code Execution

```
k.2 <- knn(train = train, test = test, cl = train_admin, k=2)

confusionMatrix(k.2, test_admin)

k.3 <- knn(train = train, test = test, cl = train_admin, k=3)

k.4 <- knn(train = train, test = test, cl = train_admin, k=4)

predict(k.4)

confusionMatrix(k.3, test_admin)

confusionMatrix(k.4, test_admin)
```

E1: Accuracy and AUC

The accuracy of my model ended up being 99.9%. Out of 2000 observations, there were only 3 misclassifications. I also calculated the AUC of my model, which was 0.999. The closer the AUC is to 1, the more accurate the model is, so this also indicated that my model is extremely accurate.

E2: Results and Implications

Overall, my model ended up being extremely accurate. The only thing that could be improved would be that emergency is over predicted. Other than that, this model is nearly perfect.

E3: Limitation

One limitation of this model is that the initial admission variable is not uniform. There are nearly as many emergency admissions as elective and observation admissions combined. This is probably why the 3 incorrect classifications were predicted to be emergency observations.

E4: Course of Action

I believe the organization can really benefit from this model. I think one thing the organization should do is to try to lower the amount of emergency admissions. I would guess that emergency admissions are more expensive to the patient than elective or observation admissions are. If we can predict a customer will be an emergency admission, we could try to get ahead of it and try to schedule more frequent check-ups to lessen the risk of an emergency admission.

F: Panopto Demonstration

Video is included in the submission and can be found using the link below.

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=45928841-b0d4-4eb7-9e39-af8e0037f248

G: Sources of Third-Party Code

knn: k-Nearest Neighbor Classification. RDocumentation. Retrieved 1/15/23. https://www.rdocumentation.org/packages/class/versions/7.3-20/topics/knn

R ifelse() Function. DataMentor. Retrieved 1/17/23. https://www.datamentor.io/r-programming/ifelse-function/

multiclass.roc: Multi-class AUC. RDocumentation. Retrieved 1/17/23. https://www.rdocumentation.org/packages/pROC/versions/1.18.0/topics/multiclass.roc

H: Sources

Grant, Peter. *Introducing k-Nearest Neighbors.* Towards Data Science. July 21, 2019. Retrieved 1/17/23. https://towardsdatascience.com/introducing-k-nearest-neighbors-7bcd10f938c5

Narkhede, Sarang. *Understanding AUC-ROC Curve*. Towards Data Science. June 26, 2018. Retreived 1/17/23. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5