**Performance Assessment: Task 2**

**A1. Proposal of Question** My research question for this performance assessment is, “Can the random forest model be used to predict the number of days a patient stays in the hospital during an initial visit?”

**A2. Defined Goal**

The goal of this analysis is to develop a random forest predictive model to help the hospital predict the number of initial days for a patient based on the medical data available.

**B1. Explanation of Prediction**

The random forest predictive method selects random samples from the dataset and creates a decision tree for each sample. A predicted result is made from each decision tree and results are averaged to create one final prediction. The expected outcome of this analysis is that my model will be able to accurately predict the number of initial days for hospital visits while accounting for overfitting.

**B2. Summary of Method Assumption**

One assumption of the random forest predictive model is that each of the decision trees in the random forest are independent from the others. This is necessary for the model to assess patterns that may exist in the data.

**B3. Packages or Libraries List**

For this assignment, I chose to use the R coding language via R Studio. The *readr* package was used to load the data into the data frame. The *dplyr* package was used for the glimpse function, which allowed me to get a quick summary of the full data set. The *corrr* package was used to visualize the correlation between my independent variables. The *ggplot2* package was also used for correlation. The caret package was used to split the data. The *randomForest* package was used to build the random forest model. Lastly, the *Metrics* package was used to calculate the RSME & MSE.

**C1. Data Preprocessing**

One data preprocessing goal that was used for this analysis was to encode categorical variables to numeric. This is a necessary step for analysis as random forest requires numeric values. Since the Complication\_risk and Asthma fields were ordinal, I was able to ordinally encode the values and dummy variables were not necessary.

**C2. Data Set Variables**

|  |  |
| --- | --- |
| Initial\_days | Numeric |
| Children | Numeric |
| Age | Numeric |
| Income | Numeric |
| VitD\_levels | Numeric |
| Doc\_visits | Numeric |
| Full\_meals\_eaten | Numeric |
| TotalCharge | Numeric |
| Additional\_charges | Numeric |
| Timely\_treatment | Numeric |
| Active\_listening | Numeric |
| Complication\_risk | Categorical |
| Asthma | Categorical |

**C3. Steps for Analysis**

The first step in preparing the data was to import the necessary libraries for analysis. Next, I loaded the raw medical data set into the data frame. I chose to rename the survey response variables for easier understanding. I assessed for duplicates, nulls. I assessed for outliers using histograms. After that, I encoded the categorical variables in my initial data set to numeric values. I created a new data frame with the initial variables I would be using in my analysis. I created a visualization of the correlation between all the variables in my analysis. Finally, I exported the cleaned data set to a CSV file.

#Import necessary libraries

library(readr) #import data

library(dplyr) #data manipulation

library(corrr) #heat map

library(ggplot2) #heat map

library(caret) #split data

library(randomForest) #build random forest model

library(Metrics) #calculate RSME & MSE

#Load the data

med <- read\_csv("WGU/D209\_medical\_raw.csv")

# Get summary of data

str(med)

#Rename survey response variables

colnames(med)[colnames(med) == 'Item1'] <- 'Timely\_admission'

colnames(med)[colnames(med) == 'Item2'] <- 'Timely\_treatment'

colnames(med)[colnames(med) == 'Item3'] <- 'Timely\_visits'

colnames(med)[colnames(med) == 'Item4'] <- 'Reliability'

colnames(med)[colnames(med) == 'Item5'] <- 'Options'

colnames(med)[colnames(med) == 'Item6'] <- 'Hours\_of\_treatment'

colnames(med)[colnames(med) == 'Item7'] <- 'Courteous\_staff'

colnames(med)[colnames(med) == 'Item8'] <- 'Active\_listening'

#Assess for duplicates

sum(duplicated(med))

#Assess for nulls

sum(is.na(med))

#Assess for outliers with histograms for quantitative variables

par(mar=c(1,1,1,1))

hist(med$Initial\_days)

hist(med$Children)

hist(med$Age)

hist(med$Income)

hist(med$VitD\_levels)

hist(med$Doc\_visits)

hist(med$Full\_meals\_eaten)

hist(med$TotalCharge)

hist(med$Additional\_charges)

hist(med$Timely\_treatment)

hist(med$Active\_listening)

#Encode categorical variables (Complication risk & Asthma) as numeric

med$Complication\_risk[med$Complication\_risk == 'Low'] <- 1

med$Complication\_risk[med$Complication\_risk == 'Medium'] <- 2

med$Complication\_risk[med$Complication\_risk == 'High'] <- 3

med$Asthma[med$Asthma == 'Yes'] <- 1

med$Asthma[med$Asthma == 'No'] <- 0

#New data frame with random forest variables, as numeric

med\_clean <- med %>%

select(Initial\_days, Children, Age, Income, VitD\_levels,

Doc\_visits, Full\_meals\_eaten, TotalCharge,

Additional\_charges, Timely\_treatment, Active\_listening,

Complication\_risk, Asthma) %>%

mutate\_at(vars(Initial\_days, Children, Age, Income, VitD\_levels,

Doc\_visits, Full\_meals\_eaten, TotalCharge,

Additional\_charges, Timely\_treatment, Active\_listening,

Complication\_risk, Asthma), as.numeric)

glimpse(med\_clean)

#Visualize correlation

med\_clean %>%

dplyr::select(where(is.numeric)) %>%

correlate() %>%

shave() %>%

rplot(print\_cor = TRUE) +

theme(axis.text.x = element\_text(angle = 90, hjust = 1))

A screenshot of a graph

Description automatically generated

#Export cleaned data set

write\_csv(med\_clean, "WGU/D209\_T2\_MV\_clean.csv") **C4. Cleaned Data Set**

See ‘D209\_T2\_MV\_clean’ file for cleaned data set.

**D1. Splitting the Data**

See below for data split code:

#Set seed to reproduce same split

set.seed(123)

#Create indices

train\_index <- createDataPartition(med\_clean$Initial\_days, p=0.8, list=FALSE)

#Create the training and testing data sets

train\_set <-med\_clean[train\_index, ]

test\_set <- med\_clean[-train\_index, ]

#Confirm\_split

dim(train\_set)

dim(test\_set)

#Further split data (y as data frame so I can export to CSV)

x\_train <- train\_set[, -which(names(train\_set) %in% "Initial\_days")]

y\_train <- as.data.frame(train\_set$Initial\_days)

x\_test <- test\_set[, -which(names(test\_set) %in% "Initial\_days")]

y\_test <- as.data.frame(test\_set$Initial\_days)

#Export as training & testing data

write\_csv(x\_train, "WGU/D209\_T2\_MV\_xtrain.csv")

write\_csv(x\_test, "WGU/D209\_T2\_MV\_xtest.csv")

write\_csv(y\_train, "WGU/D209\_T2\_MV\_ytrain.csv")

write\_csv(y\_test, "WGU/D209\_T2\_MV\_ytest.csv")

See the following files for training and test data sets:

1. ‘D209\_T2\_MV\_xtrain’
2. ‘D209\_T2\_MV\_xtest’
3. ‘D209\_T2\_MV\_ytrain’
4. ‘D209\_T2\_MV\_ytest’

**D2. Output and Intermediate Calculations**

I did not perform any intermediate calculations in this analysis.

**D3. Code Execution**

See below for random forest model code:

# Convert y back to vector

y\_train <- as.vector(unlist(y\_train))

y\_test <- as.vector(unlist(y\_test))

#Train random forest model on training set

rf\_model <- randomForest(y\_train ~ ., data=x\_train, ntree=500)

#Print summary

print(rf\_model)

#Make predictions on the testing set

rf\_predictions <- predict(rf\_model, newdata = x\_test)

#Calculate r-squared

ss\_res <- sum((y\_test - rf\_predictions) ^2)

ss\_tot <- sum((y\_test - mean(y\_test)) ^2)

r\_squared <- 1 - (ss\_res / ss\_tot)

print(r\_squared)



#RMSE & MSE

rmse <- rmse(y\_test, rf\_predictions)

print(rmse)



mse <- rmse^2

print(mse)



#Range of Initial days

range\_dependent <- range(med\_clean$Initial\_days)

print(range\_dependent)



**E1. Accuracy and MSE**

The accuracy of my random forest model was assessed by examining the RMSE, MSE, and r-squared value. The MSE of my model was 9.65. The RMSE of my model was 3.11. This can be interpreted as: my model’s prediction are 3.11 units away from the actual value. The r-squared of my model was 0.99

**E2. Results and Implications**

The high r-squared value suggests that my model fits the data almost perfectly. Since my range of initial days was 1 to 72, I would consider the RMSE margin of error to be reasonable. In conclusion, initial days can be accurately predicted using the variables included in my random forest model.

**E3. Limitation**

One possible limitation of this analysis is that the random forest method is susceptible to overfitting. This can lead to poor performance in real world applications.

**E4. Course of Action**

I recommended expanding the analysis to include all variables in the data set, which may lead to a different set of final variables. This could lead to an even lower RSME, therefore a more accurate model.

**F. Panopto Recording**

Link to video included with submission.

**G. Sources for Third-Party Code**

**H. Sources**