**Random Forest Predictions**

By Dante Navarro and Derek Chang

**Predicting the Length of Stay Based on the First 3 Days of Data**

\*\*Patients with a Length of Stay (time from transfer from ICU to Discharge) of < 3 were omitted

y - days on unit after day 3 = (lengthOfStay - 3)

x1 - hours to transfer = (admission2ICU + icuDay)

x2 - day of week transferred = date\_transfer

x3 - # walks on day 1 = (iterate through and sum)

x4 - dist walk day 1 = (iterate through and sum)

x5 - # walks on day 2 = (iterate through and sum)

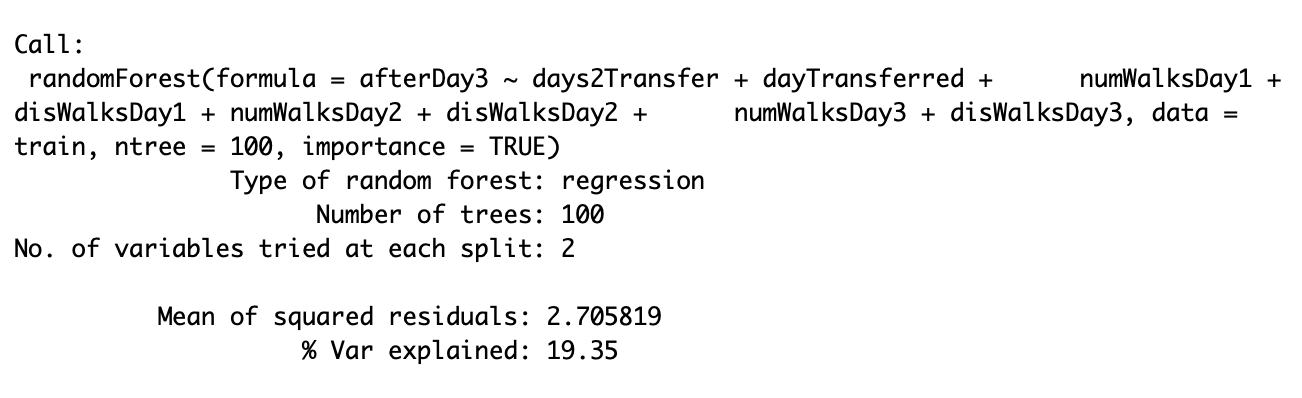
x6- dist walk day 2 = (iterate through and sum)

x7 - # walks on day 3 = (iterate through and sum)

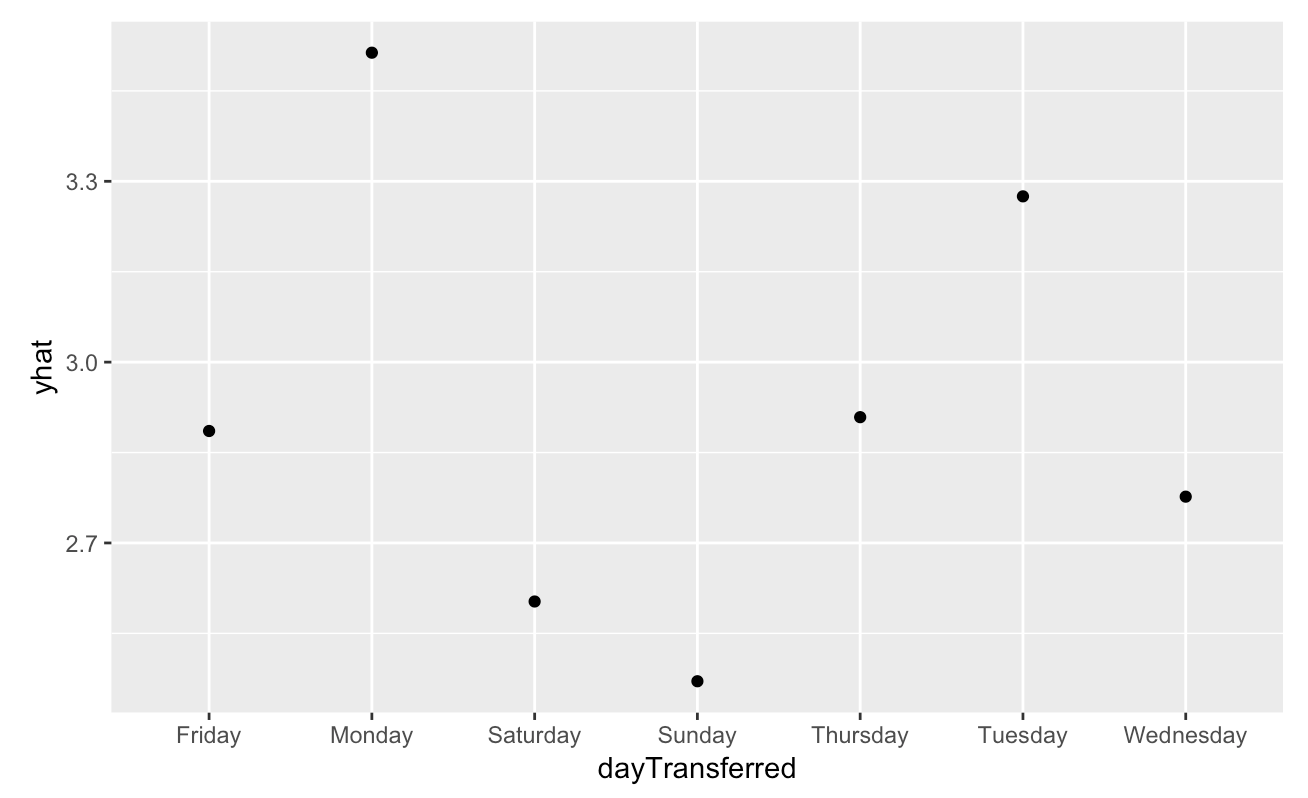
x8- dist walk day 3 = (iterate through and sum)

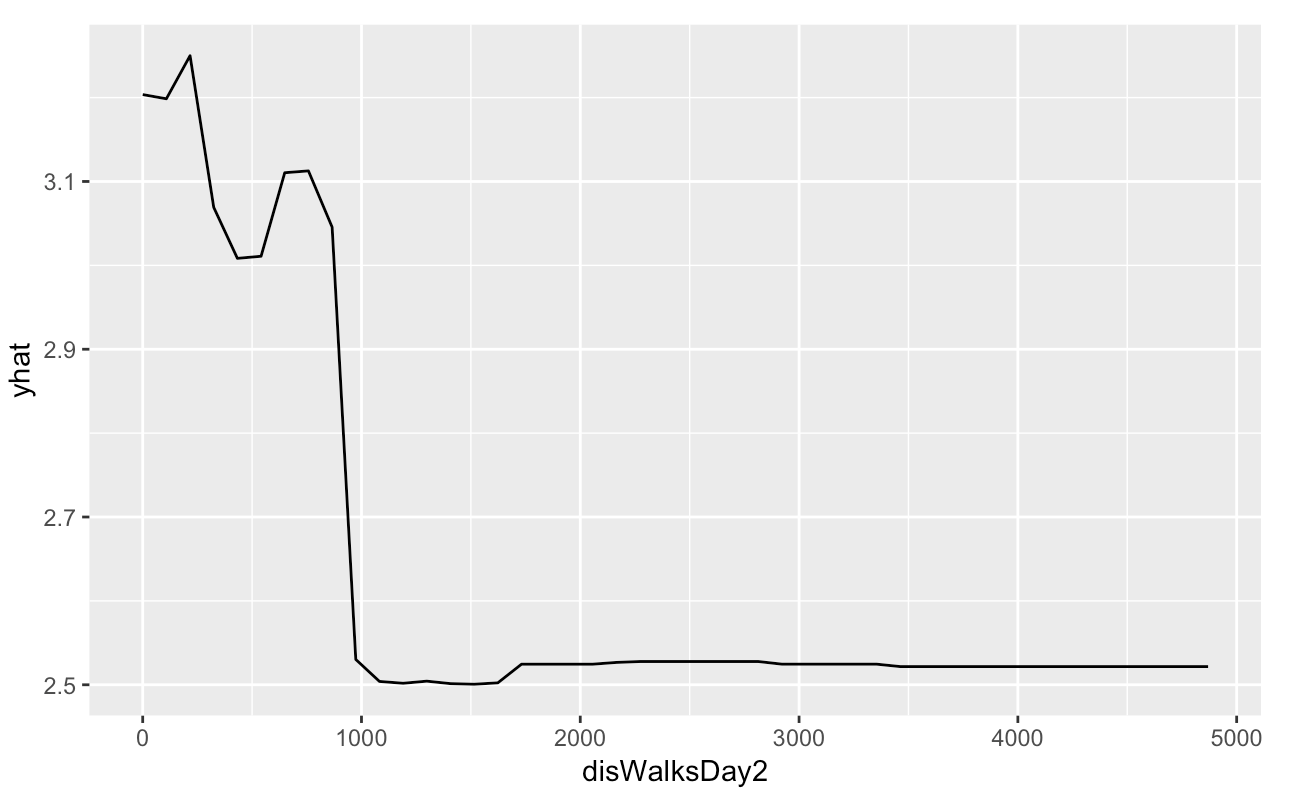
*Random Forest Output*

*A screenshot of a social media post

Description automatically generatedVariable Importance Plot*

*Partial Regression Plot*

*dayTransferred*

*disWalksDay2*

*dayTransferred, disWalksDay2*

*A close up of a map

Description automatically generated*

**Predicting Chance Of Walking Each Hour**

Y = Did the patient walk (yes or no)

X1 = Day On Unit

X2 = Hour Of Stay

X3 = Ambulations

X4 = Daily Ambulation Total

X5 = Hour (of the day)

X6 = Number Of Times (walks in the hour)

X7 = Hours Since Last Walk

X8 = Distance Of Last Walk

**Random Forest Code:**

library(randomForest)

library(pdp)

library(ggplot2)

library(rpart)

ambData <- read.csv("Patients\_hourly\_data.csv",

fileEncoding="UTF-8-BOM",

header = TRUE)

set.seed(100)

trainRowCount <- floor(0.7 \* nrow(ambData))

trainIndex <- sample(nrow(ambData), trainRowCount)

train <- ambData[trainIndex,]

test <- ambData[-trainIndex,]

predictorColumns <- c("Day.On.Unit", "Hour.Of.Stay", "Ambulations", "Daily.Ambulation.Total",

"Hour", "Number.Of.Times", "Hours.Since.Last.Walk", "Distance.Of.Last.Walk", "Total.Distance.Walked")

rf <- randomForest(Walking ~ Day.On.Unit + Hour.Of.Stay + Ambulations +

Daily.Ambulation.Total + Hour + Number.Of.Times + Hours.Since.Last.Walk + Distance.Of.Last.Walk + Total.Distance.Walked, data = train, ntree=100, importance = TRUE)

predictions <- predict(rf, test[predictorColumns])

print(rf)

varImpPlot(rf,type=2)

par <- partial(rf, pred.var = c("Day.On.Unit"), chull = TRUE)

plot <- autoplot(par, contour = TRUE)

print(plot)

**Random Forest Output:**

Call:

randomForest(formula = Walking ~ Day.On.Unit + Hour.Of.Stay + Ambulations + Daily.Ambulation.Total + Hour + Number.Of.Times + Hours.Since.Last.Walk + Distance.Of.Last.Walk + Total.Distance.Walked, data = train, ntree = 100, importance = TRUE)

Type of random forest: classification

Number of trees: 100

No. of variables tried at each split: 3

OOB estimate of error rate: 0%

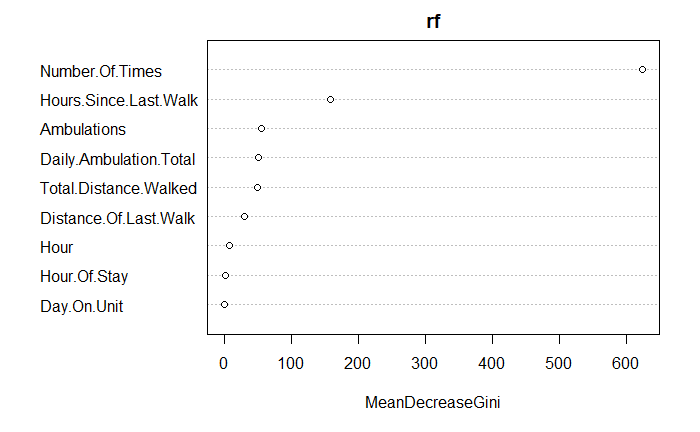
Confusion matrix:

no yes class.error

no 7063 0 0

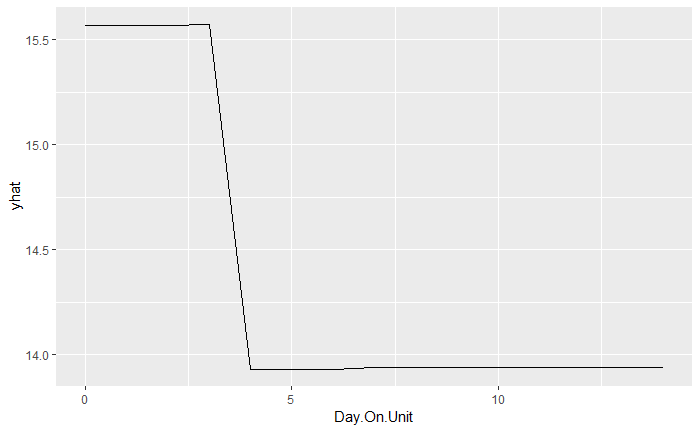
yes 0 521 0

**Variable Importance Plot:**



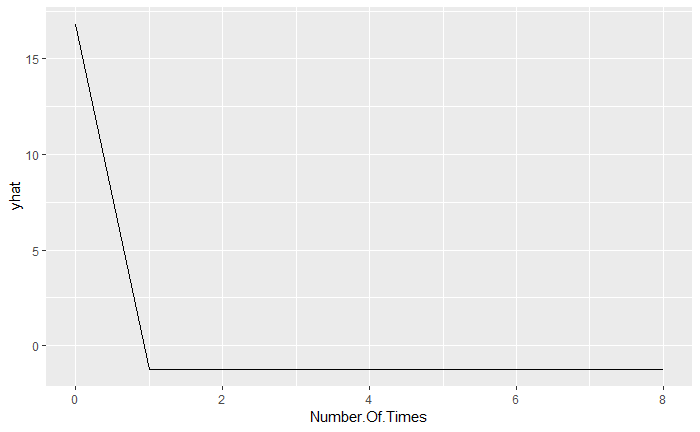
**Partial Regression Plots:**

par <- partial(rf, pred.var = c("Day.On.Unit"), chull = TRUE)

plot <- autoplot(par, contour = TRUE)

par <- partial(rf, pred.var = c("Number.Of.Times"), chull = TRUE)

plot <- autoplot(par, contour = TRUE)



par <- partial(rf, pred.var = c("Number.Of.Times", "Day.On.Unit"), chull = TRUE)

plot <- autoplot(par, contour = TRUE)

