analyze.R

dano

Wed Nov 11 14:47:02 2015

#Use this to plot and analyze   
library(ggplot2)  
library(corrplot)  
  
plot\_omnipod\_type\_frequency <- function(data){  
 type\_frequency <- data %>%  
 group\_by(type) %>%  
 summarize(type\_count = n())  
   
 plot <- ggplot(aes(x = type, y = type\_count), data = type\_frequency) +  
 geom\_bar(stat = "identity") +  
 theme(axis.text.x=element\_text(angle = -45, hjust = 0)) +  
 xlab("Record Type") +  
 ylab("Record Type Frequency") +  
 ggtitle("Omnipod Record Type Frequency")  
   
 print(plot)  
}  
  
hist\_bolus <- function(omni\_pod){  
 data <- filter(omni\_pod,  
 type == "Bolus Insulin")  
   
 hist(data$value, xlab = "Bolus Size (units)", ylab = "Frequency", main = "Hist of Bolus sizes")  
}  
  
hist\_fitbit <- function(fitbit){  
 hist(fitbit$floors, xlab = "Floors", main = "Hist of Fitbit Floors")  
 hist(fitbit$distance, xlab = "Distance", main = "Hist of Fitbit Distance")  
 hist(fitbit$calories, xlab = "Calories", main = "Hist of Fitbit Calories")  
 hist(fitbit$steps, xlab = "Steps", main = "Hist of Fitbit Steps")  
 hist(fitbit$elevation, xlab = "Elevation", main = "Hist of Fitbit Elevation")  
}  
  
corrplot\_fitbit <- function(fitbit){  
 old <- par()  
 corrplot(cor(select(fitbit, -date, -time)))  
 par(old)  
}  
  
#LDA on Glucose levels in comparission to true/false  
library(MASS)  
library(lubridate)  
library(dplyr)

##   
## Attaching package: 'dplyr'  
##   
## The following objects are masked from 'package:lubridate':  
##   
## intersect, setdiff, union  
##   
## The following object is masked from 'package:MASS':  
##   
## select  
##   
## The following objects are masked from 'package:stats':  
##   
## filter, lag  
##   
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

data\_set=read.csv("data\_set.csv", stringsAsFactors = F)%>%  
 select(-X) %>%  
 mutate(datetime = ymd\_hms(datetime)) %>%  
 rename(basal\_delivery = basal\_delievery)  
newdata= data\_set %>% select(-datetime, -basal\_insulin\_hourly\_rate, -bolus\_insulin\_units, -glucose, -meal\_approx\_carbs, -low\_bg, -in\_range\_bg, -after\_noon, -dawn\_phenomenon)  
newdata = newdata[complete.cases(newdata),]  
head(newdata)

## basal\_delivery bolus\_burndown total\_insulin\_burndown acting\_carbs  
## 37 0.04166667 5.461270 7.571352 58.99243  
## 38 0.04166667 5.352045 7.461592 57.81258  
## 39 0.04166667 5.245004 7.354026 56.65633  
## 40 0.04166667 5.140104 7.248613 55.52321  
## 41 0.04166667 5.037302 7.145307 54.41274  
## 42 0.04166667 4.936556 7.044068 53.32449  
## predicted\_bg floors distance calories steps elevation  
## 37 82 0 0.0003790364 5.3875 1 0  
## 38 82 0 0.0000000000 5.3875 0 0  
## 39 85 0 0.0000000000 5.3875 0 0  
## 40 88 0 0.0000000000 6.8960 0 0  
## 41 91 0 0.0079597649 8.4045 21 0  
## 42 91 0 0.0102339834 11.6370 27 0  
## predicted\_bg\_lag\_1hour predicted\_bg\_lag\_2hour predicted\_bg\_lag\_3hour  
## 37 128 170 217  
## 38 120 166 216  
## 39 115 155 216  
## 40 108 149 210  
## 41 105 146 208  
## 42 103 150 203  
## predicted\_bg\_lead\_1hour predicted\_bg\_lead\_2hour predicted\_bg\_lead\_3hour  
## 37 117 146 178  
## 38 125 149 179  
## 39 126 151 180  
## 40 126 153 182  
## 41 126 154 182  
## 42 128 156 182  
## predicted\_bg\_lead\_4hour predicted\_bg\_lead\_5hour  
## 37 178 165  
## 38 176 165  
## 39 175 167  
## 40 174 166  
## 41 174 164  
## 42 173 163  
## total\_insulin\_burndown\_lag\_1hour total\_insulin\_burndown\_lag\_2hour  
## 37 9.076964 6.480157  
## 38 8.937091 6.392220  
## 39 8.800016 6.306042  
## 40 8.665682 6.221588  
## 41 8.534035 6.138823  
## 42 8.405021 6.057713  
## total\_insulin\_burndown\_lag\_3hour total\_insulin\_burndown\_lead\_1hour  
## 37 7.686404 7.275716  
## 38 7.574343 7.171868  
## 39 7.464522 7.070097  
## 40 7.356899 6.970362  
## 41 7.251427 6.872622  
## 42 7.148065 6.776836  
## total\_insulin\_burndown\_lead\_2hour total\_insulin\_burndown\_lead\_3hour  
## 37 6.157883 5.280700  
## 38 6.076392 5.216753  
## 39 5.996530 5.154085  
## 40 5.918267 5.092670  
## 41 5.841568 5.032483  
## 42 5.766403 4.973500  
## total\_insulin\_burndown\_lead\_4hour total\_insulin\_burndown\_lead\_5hour  
## 37 4.592361 4.052209  
## 38 4.542180 4.012832  
## 39 4.493003 3.974242  
## 40 4.444810 3.936423  
## 41 4.397580 3.899362  
## 42 4.351295 3.863041  
## acting\_carbs\_lag\_1hour acting\_carbs\_lag\_2hour acting\_carbs\_lag\_3hour  
## 37 75.17673 39.35765 50.15523  
## 38 73.67319 38.57050 49.15213  
## 39 72.19973 37.79909 48.16909  
## 40 70.75573 37.04311 47.20570  
## 41 69.34062 36.30224 46.26159  
## 42 67.95381 35.57620 45.33636  
## acting\_carbs\_lead\_1hour acting\_carbs\_lead\_2hour acting\_carbs\_lead\_3hour  
## 37 46.29235 36.32638 28.50592  
## 38 45.36650 35.59985 27.93580  
## 39 44.45917 34.88786 27.37708  
## 40 43.56999 34.19010 26.82954  
## 41 42.69859 33.50630 26.29295  
## 42 41.84462 32.83617 25.76709  
## acting\_carbs\_lead\_4hour acting\_carbs\_lead\_5hour calories\_sum\_past\_1hour  
## 37 22.36907 17.55338 96.7595  
## 38 21.92169 17.20232 93.5270  
## 39 21.48326 16.85827 91.8030  
## 40 21.05359 16.52110 93.3115  
## 41 20.63252 16.19068 96.3285  
## 42 20.21987 15.86687 101.0695  
## calories\_sum\_past\_2hour calories\_sum\_past\_3hour calories\_sum\_next\_1hour  
## 37 195.0275 284.2445 77.1490  
## 38 187.4850 278.6415 77.1490  
## 39 185.9765 273.2540 78.8730  
## 40 187.4850 271.5300 78.8730  
## 41 190.5020 273.0385 77.3645  
## 42 195.0275 277.5640 74.3475  
## calories\_sum\_next\_2hour calories\_sum\_next\_3hour calories\_sum\_next\_4hour  
## 37 156.6685 222.8352 287.4972  
## 38 156.6685 222.8362 287.4982  
## 39 156.6685 222.8372 287.4992  
## 40 158.1770 222.8382 287.5002  
## 41 156.6687 221.3307 285.9927  
## 42 153.6527 218.3147 282.9767  
## calories\_sum\_next\_5hour steps\_sum\_past\_1hour steps\_sum\_past\_2hour  
## 37 353.6680 262 405  
## 38 353.6690 240 405  
## 39 353.6700 240 394  
## 40 353.6710 240 394  
## 41 352.1635 261 415  
## 42 349.1475 288 423  
## steps\_sum\_past\_3hour steps\_sum\_next\_1hour steps\_sum\_next\_2hour  
## 37 540 49 109  
## 38 505 48 108  
## 39 456 48 108  
## 40 443 48 108  
## 41 464 48 108  
## 42 484 27 87  
## steps\_sum\_next\_3hour steps\_sum\_next\_4hour steps\_sum\_next\_5hour  
## 37 109 109 109  
## 38 108 108 108  
## 39 108 108 108  
## 40 108 108 108  
## 41 108 108 108  
## 42 87 87 87  
## floors\_sum\_past\_1hour floors\_sum\_past\_2hour floors\_sum\_past\_3hour  
## 37 0 0 0  
## 38 0 0 0  
## 39 0 0 0  
## 40 0 0 0  
## 41 0 0 0  
## 42 0 0 0  
## floors\_sum\_next\_1hour floors\_sum\_next\_2hour floors\_sum\_next\_3hour  
## 37 0 0 0  
## 38 0 0 0  
## 39 0 0 0  
## 40 0 0 0  
## 41 0 0 0  
## 42 0 0 0  
## floors\_sum\_next\_4hour floors\_sum\_next\_5hour high\_bg  
## 37 0 0 FALSE  
## 38 0 0 FALSE  
## 39 0 0 FALSE  
## 40 0 0 FALSE  
## 41 0 0 FALSE  
## 42 0 0 FALSE

#head(newdata)  
try1= lda(high\_bg ~ ., data = newdata)

## Warning in lda.default(x, grouping, ...): variables are collinear

pred = predict(try1,newdata)  
names(pred)

## [1] "class" "posterior" "x"

table(newdata$high\_bg,pred$class)

##   
## FALSE TRUE  
## FALSE 4437 25  
## TRUE 147 760

vlda = function(v,formula,data,cl){  
 require(MASS)  
 grps = cut(1:nrow(data),v,labels=FALSE)[sample(1:nrow(data))]  
 pred = lapply(1:v,function(i,formula,data){  
 omit = which(grps == i)  
 z = lda(formula,data=data[-omit,])  
 predict(z,data[omit,])  
 },formula,data)  
   
 wh = unlist(lapply(pred,function(pp)pp$class))  
 table(wh,cl[order(grps)])  
}  
tt = vlda(5, high\_bg~., newdata, newdata$high\_bg)

## Warning in lda.default(x, grouping, ...): variables are collinear

## Warning in lda.default(x, grouping, ...): variables are collinear

## Warning in lda.default(x, grouping, ...): variables are collinear

## Warning in lda.default(x, grouping, ...): variables are collinear

## Warning in lda.default(x, grouping, ...): variables are collinear

tt

##   
## wh FALSE TRUE  
## FALSE 4431 151  
## TRUE 31 756

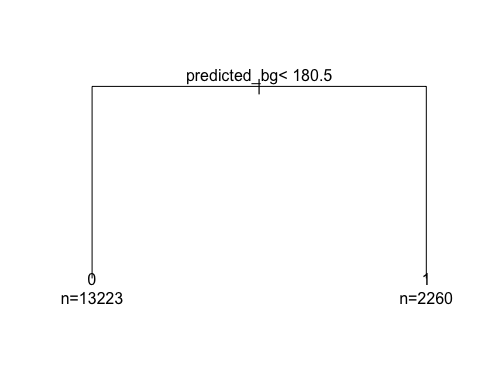
error = sum(tt[row(tt) != col(tt)])/ sum(tt)  
error

## [1] 0.03389831

library(rpart)  
glucose.rpart = rpart(high\_bg~., data = data\_set)  
glucose.rpart

## n=15483 (4741 observations deleted due to missingness)  
##   
## node), split, n, deviance, yval  
## \* denotes terminal node  
##   
## 1) root 15483 1930.116 0.1459665   
## 2) predicted\_bg< 180.5 13223 0.000 0.0000000 \*  
## 3) predicted\_bg>=180.5 2260 0.000 1.0000000 \*

plot(glucose.rpart)  
text(glucose.rpart, use.n=TRUE,xpd=TRUE)



plot(try1)

