

# Feature Selection

Christoph Völtzke

2023

## Packages

```
# Processing  
library(readr) # read data  
library(tidyverse) # data manipulation  
# Feature Selection and Imbalance  
library(ggcorrplot) # features correlation  
library(caret) # find correlation function  
library(Boruta) # Boruta feature selection  
library(infotheo) # Mutual information feature importance
```

## Information

- In this document first the combined data set from pre-processing is further cleaned for the purposes of this study.
- Then data is split to only perform any analyses on the training data, but not on the test data.
- Next feature selection is performed. All steps are explained below in detail.
  - Included are: Correlation checks, Mutual Information criteria, Boruta feature selection and expert knowledge.
- Plots are generated used in the report and dfs are created and saved which are used for model training

## Load data sets which need to be balanced

```
source("Functions/data_split.R")
```

## Original data

```
full_data <- read_csv("~/GitHub/Sweet_Dreams_Ahead_Machine_Learning_Models_for_Nocturnal_Hypoglycemia_P  
  
## Rows: 833 Columns: 105  
## -- Column specification -----  
## Delimiter: ","  
## chr    (2): wday, weekend
```

```
## dbl (103): periods, patient, hour_0_1_before_mean_glucose_ti, hours_1_2_befo...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

## Cleaning of full data sets for different tasks (Hyper/Hypo) w/o Outcomes

Only relevant variables are selected, variables are renamed to make them easier to understand for the reader, Two data conditions are subsetted. Full data condition and Lifestyle condition

```
# cleaning of unnecessary vars
data <- full_data %>%
  mutate(wday = factor(wday),
         weekend = factor(weekend)) %>%
  select(-periods,-DM_type,-Night_time_mean_glucose_ti,-Night_time_sd_glucose_ti,
        -Night_time_min_glucose_ti,-Night_time_max_glucose_ti,-Night_late_mean_glucose,
        -Night_late_sd_glucose,-Night_late_min_glucose,-Night_late_max_glucose,
        -Night_late_Hyper_events,-Night_late_Hypo_events,-Night_early_Night_class_hyper_out,
        -Night_late_Night_class_hyper_out,-Night_class_hyper_out,-Day_Night_class_hypo_out,
        -Day_Night_class_hyper_out,-Day_Night_class_hyper_30) %>%
  na.omit()

# cleaning for full night
full_data <- data %>%
  mutate(Night_class_hyper_30 = factor(Night_class_hyper_30),
         Night_class_hypo_out = factor(Night_class_hypo_out)) %>%
  select(-Night_early_mean_glucose,-Night_early_sd_glucose,-Night_early_min_glucose,
        -Night_early_max_glucose,-Night_early_Hyper_events,-Night_early_Hypo_events,
        -Night_early_Night_class_hypo_out,-Night_late_Night_class_hypo_out,
        -Night_early_Night_class_hyper_30,-Night_late_Night_class_hyper_30) %>%
  select(Night_class_hypo_out,Night_class_hyper_30,patient,
        hour_0_1_before_mean_glucose_ti:slope_day,COB,Sum_carbs,Sum_calories,
        Sum_Fat,Num_Meals,Last_meal,AOB,Sum_steps,wday,weekend,Geslacht:dosA10BB)

# Features are renamed to make them easier to understand in the
# study as some names will be depicted in one of the tables
full_data <- full_data %>%
  rename(mean_0_1 = hour_0_1_before_mean_glucose_ti, mean_1_2 =hours_1_2_before_mean_glucose_ti,
        mean_2_3 =hours_2_3_before_mean_glucose_ti, mean_3_4 =hours_3_4_before_mean_glucose_ti,
        mean_4_5 =hours_4_5_before_mean_glucose_ti, mean_5_6 =hours_5_6_before_mean_glucose_ti,
        mean_6pl =hours_7_plus_mean_glucose_ti,
        sd_0_1 = hour_0_1_before_sd_glucose_ti, sd_1_2 =hours_1_2_before_sd_glucose_ti,
        sd_2_3 =hours_2_3_before_sd_glucose_ti, sd_3_4 =hours_3_4_before_sd_glucose_ti,
        sd_4_5 =hours_4_5_before_sd_glucose_ti,sd_5_6 =hours_5_6_before_sd_glucose_ti,
        sd_6pl =hours_7_plus_sd_glucose_ti,
        min_0_1 = hour_0_1_before_min_glucose_ti, min_1_2 =hours_1_2_before_min_glucose_ti,
        min_2_3 =hours_2_3_before_min_glucose_ti, min_3_4 =hours_3_4_before_min_glucose_ti,
        min_4_5 =hours_4_5_before_min_glucose_ti,min_5_6 =hours_5_6_before_min_glucose_ti,
        min_6pl =hours_7_plus_min_glucose_ti,
        max_0_1 = hour_0_1_before_max_glucose_ti, max_1_2 =hours_1_2_before_max_glucose_ti,
        max_2_3 =hours_2_3_before_max_glucose_ti, max_3_4 =hours_3_4_before_max_glucose_ti,
        max_4_5 =hours_4_5_before_max_glucose_ti,max_5_6 =hours_5_6_before_max_glucose_ti,
        max_6pl =hours_7_plus_max_glucose_ti,
```

```

mean_day = Day_mean_glucose, sd_day = Day_sd_glucose, min_day = Day_min_glucose,
max_day = Day_max_glucose, Hyper_n = Day_Hyper_events, Hypo_n = Day_Hypo_events,
hyper_Ind = hyper_index, hypo_Ind = hypo_index, tir_70_180 = in_range_70_180,
tir_63_140 = in_range_63_140, slope_0_1 = hour_0_1_before_slope_day,
slope_1_2 = hours_1_2_before_slope_day, Steps = Sum_steps,
Carbs = Sum_carbs, Calories = Sum_calories, Fat = Sum_Fat,
Height = Lengte_poli1, Weight = Gewicht_poli1, Gender = Geslacht, Age = Leeftijd_poli1,
T2D_Dur = Diabetes_Length, T2D_Start = Leeftijd_aanvangDM2, HbA1c = SerumHbA1c_1,
insulin = insulin_bin, fast_ins = atcA10AB, dos_fast = dosA10AB, int_long = atcA10AD,
dos_int_long = dosA10AD, long_ins = atcA10AE, dos_long = dosA10AE, SU_der = atcA10BB,
dos_SU = dosA10BB)

# LIFESTYLE DATA
full_data_life_plus <- full_data %>% select(Night_class_hypo_out, Night_class_hyper_30, patient,
COB:Last_meal, AOB:Steps, wday: dos_SU)

write.csv(full_data, "Data/full_data.csv", row.names = FALSE)

```

## Data set preparation - Selection of Outcome variable

```

full_data <- full_data %>%
  mutate(Night_class_hypo_out = as.factor(Night_class_hypo_out),
         Night_class_hyper_30 = as.factor(Night_class_hyper_30)) %>%
  select(-Night_class_hyper_30, -patient)

full_data_life_plus <- full_data_life_plus %>%
  mutate(Night_class_hypo_out = as.factor(Night_class_hypo_out),
         Night_class_hyper_30 = as.factor(Night_class_hyper_30)) %>%
  select(-Night_class_hyper_30, -patient)

```

## Data split train/test

```

full_data_split <- data_split(full_data, train_size= 0.75, seed = 123)
# the seed of the first fold is used to do the feature selection,
# In correspondence with senior researchers this approach is sufficient
full_data_train <- full_data_split[[1]]
full_data_test <- full_data_split[[2]]

full_data_life_plus_split <- data_split(full_data_life_plus, train_size= 0.75, seed = 123)
full_data_life_plus_train <- full_data_life_plus_split[[1]]
full_data_life_plus_test <- full_data_life_plus_split[[2]]

```

## Steps - Feature selection

Feature selection is performed in multiple steps.

1. Correlations were checked, but only as an optional step to see how data is distributed



```
ncol(full_data_train_col)
```

```
## [1] 61
```

```
# Very high/Low correlations are detected for the CGM features  
# However, this is to be expected due to the same data origin for feature extration
```

```
cor_matrix_dv <- cor_matrix %>%  
  filter(cor_matrix$Night_class_hypo_out <= -0.5 | cor_matrix$Night_class_hypo_out >= 0.5)  
cor_matrix_dv
```

```
##           Night_class_hypo_out  mean_0_1  mean_1_2  mean_2_3  
## Night_class_hypo_out           1 -0.2468591 -0.1733234 -0.137744  
##           mean_3_4  mean_4_5  mean_5_6  mean_6pl  sd_0_1  
## Night_class_hypo_out -0.1453978 -0.08404654 -0.08756489 -0.2192139 0.03367274  
##           sd_1_2  sd_2_3  sd_3_4  sd_4_5  sd_5_6  
## Night_class_hypo_out 0.04037027 0.1372531 0.08963487 0.00360119 0.05243419  
##           sd_6pl  min_0_1  min_1_2  min_2_3  min_3_4  
## Night_class_hypo_out -0.09271161 -0.2585558 -0.1861614 -0.1628456 -0.1657532  
##           min_4_5  min_5_6  min_6pl  max_0_1  max_1_2  
## Night_class_hypo_out -0.08991712 -0.1050649 -0.1948245 -0.211965 -0.1436362  
##           max_2_3  max_3_4  max_4_5  max_5_6  max_6pl  
## Night_class_hypo_out -0.0923284 -0.123468 -0.06498487 -0.07042897 -0.2054673  
##           mean_day  sd_day  min_day  max_day  Hyper_n  
## Night_class_hypo_out -0.226027 0.001770546 -0.3101003 -0.1631203 -0.1738326  
##           Hypo_n tir_63_140 tir_70_180 hypo_Ind hyper_Ind  
## Night_class_hypo_out 0.3411632 0.1894531 0.09610079 0.2925526 -0.1374126  
##           GMI  GRADE  LBG1  HBGI  slope_0_1  
## Night_class_hypo_out -0.2261272 -0.1497352 0.3905704 -0.1431068 -0.09618077  
##           slope_1_2  slope_day  COB  Carbs  Calories  
## Night_class_hypo_out -0.1728225 -0.03396408 -0.02418628 -0.05013866 -0.02387719  
##           Fat  Num_Meals  Last_meal  AOB  Steps  
## Night_class_hypo_out 0.0729468 0.03203649 -0.04615708 0.05746676 -0.05884553  
##           Gender  Age  T2D_Dur  T2D_Start  Height  
## Night_class_hypo_out 0.02422907 -0.04784589 -0.004552603 0.004119618 0.09651962  
##           Weight  BMI  HbA1c  insulin  long_ins  
## Night_class_hypo_out 0.1114402 0.04331595 -0.06486763 0.04599986 0.06312614  
##           dos_long  fast_ins  dos_fast  int_long  dos_int_long  
## Night_class_hypo_out 0.01484222 0.001888556 0.0620533 -0.08466572 -0.08797856  
##           SU_der  dos_SU  
## Night_class_hypo_out 0.0006296654 0.03153161
```

```
# No variable has a correlation of 0.5/-0.5 or higher/lower with the DV
```

## Mutual information criteria

Features based on Mutual Information are compared with the outcome variable

A Threshold is applied with 0.01 for Full data condition and 0.005 for Lifestyle condition

```

options(scipen = 999)
all <- infotheo::discretize(full_data_train)
mut_inf <- mutinformation(all,method= "emp")
mut_inf <- as.data.frame(mut_inf[,1])
mut_inf <- rownames_to_column(mut_inf, var = "feature")
mut_inf <- mut_inf %>%
  filter(feature != "Night_class_hypo_out") %>%
  rename(importance = "mut_inf[, 1]") %>%
  arrange(desc(importance)) %>%
  filter(importance >= 0.01)
print(mut_inf)

```

```

##      feature importance
## 1    hypo_Ind 0.08717653
## 2    min_day 0.08204313
## 3      LBGI 0.08009280
## 4    Hypo_n 0.06674338
## 5    max_6pl 0.05382713
## 6  mean_0_1 0.05251379
## 7      GMI 0.05130710
## 8  mean_day 0.05124863
## 9  mean_6pl 0.05022811
## 10   min_6pl 0.04829738
## 11   min_0_1 0.04730316
## 12   max_0_1 0.04706128
## 13   Hyper_n 0.03484570
## 14      BMI 0.02928449
## 15   sd_6pl 0.02860616
## 16   max_day 0.02844742
## 17   min_1_2 0.02820066
## 18 hyper_Ind 0.02720341
## 19   HbA1c 0.02519141
## 20   HBGI 0.02511277
## 21   max_1_2 0.02414566
## 22   Weight 0.02327485
## 23   GRADE 0.02143922
## 24   min_3_4 0.02143409
## 25     Steps 0.02124435
## 26  mean_1_2 0.02114629
## 27   Height 0.02093765
## 28   min_2_3 0.02071133
## 29 tir_63_140 0.02005053
## 30   T2D_Dur 0.01963625
## 31  slope_1_2 0.01840997
## 32     AOB 0.01790274
## 33  mean_4_5 0.01644448
## 34     Age 0.01580270
## 35  mean_5_6 0.01521286
## 36   max_4_5 0.01513346
## 37  Calories 0.01411750
## 38  mean_2_3 0.01404905
## 39  mean_3_4 0.01362745
## 40 T2D_Start 0.01249625

```

```
## 41    max_3_4 0.01179800
## 42    max_2_3 0.01165989
## 43    min_5_6 0.01136459
## 44    min_4_5 0.01096446
```

Due to lower similarity with non CGM features a lower threshold is tested for only the variables in the Lifestyle condition. *NOTE* Values stay the same to the approach before as the mutual information approach is independent of any model and always produces the same estimates no matter how many variables are included.

```
options(scipen = 999)
life <- infotheo::discretize(full_data_life_plus_train)
mut_inf_life <- mutinformation(life, method= "emp")
mut_inf_life <- as.data.frame(mut_inf_life[,1])
mut_inf_life <- rownames_to_column(mut_inf_life, var = "feature")
mut_inf_life <- mut_inf_life %>%
  filter(feature != "Night_class_hypo_out") %>%
  rename(importance = "mut_inf_life[, 1]") %>%
  arrange(desc(importance)) %>%
  filter(importance >= 0.005)
print(mut_inf_life)
```

```
##      feature  importance
## 1      BMI 0.029284486
## 2     HbA1c 0.025191415
## 3    Weight 0.023274848
## 4     Steps 0.021244346
## 5    Height 0.020937652
## 6   T2D_Dur 0.019636253
## 7      AOB 0.017902744
## 8      Age 0.015802704
## 9   Calories 0.014117499
## 10 T2D_Start 0.012496245
## 11      Fat 0.009632974
## 12     COB 0.009140614
## 13 Num_Meals 0.007336847
## 14 dos_long 0.007323444
## 15     wday 0.006967453
## 16    Carbs 0.006807550
## 17 Last_meal 0.006467564
```

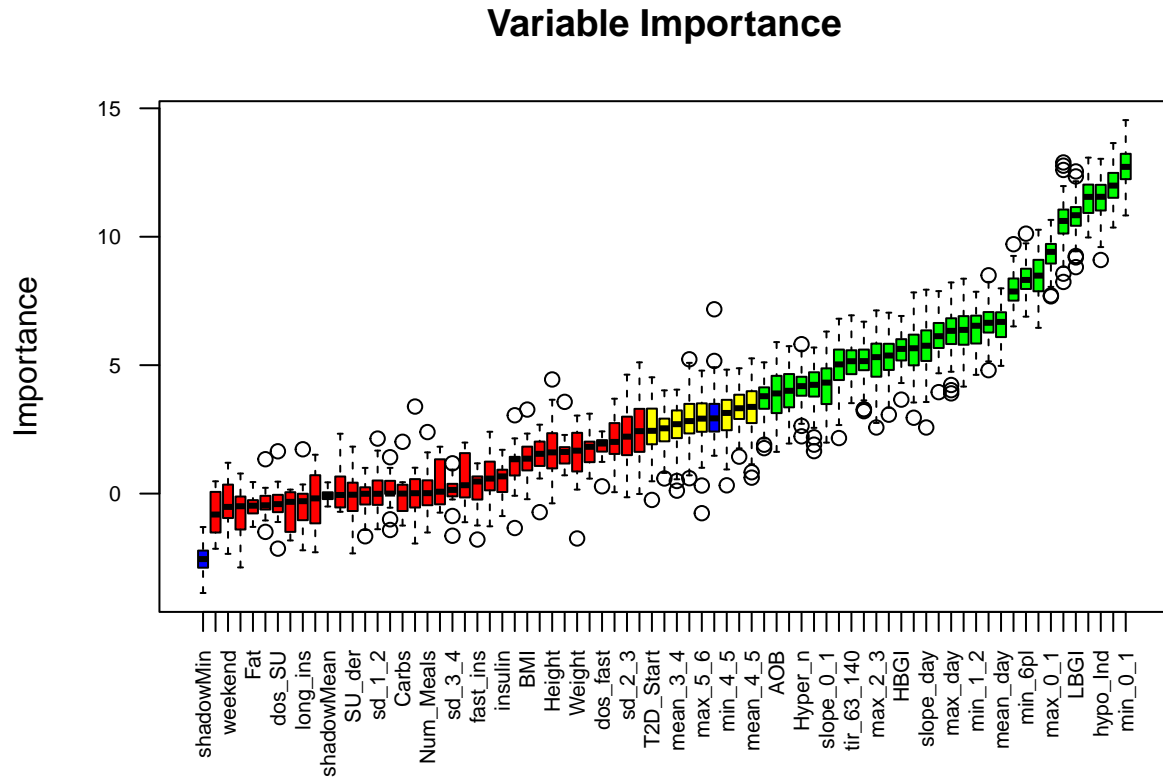
## Boruta Feature selection

### For Full data condition

Boruta feature selection for full data condition. This is repeated for Lifestyle as importance scores change due to inclusion of certain variables.

```
set.seed(123) # as Boruta is an iterative process a seed value is necessary
boruta_output_full <- Boruta(Night_class_hypo_out ~ ., data=full_data_train, doTrace=0)
boruta_signif_full <- getSelectedAttributes(boruta_output_full, withTentative = TRUE)
roughFixMod <- TentativeRoughFix(boruta_output_full)
boruta_signif <- getSelectedAttributes(roughFixMod)
```

```
# This plot shows the distribution of selected, tentative and
# rejected variables for the full data condition
imps_full <- attStats(boruta_output_full)
imps_full = imps_full[imps_full$decision != 'Rejected', c('meanImp', 'decision')]
plot(boruta_output_full, cex.axis=.7, las=2, xlab="", main="Variable Importance")
```



## For Lifestyle condition

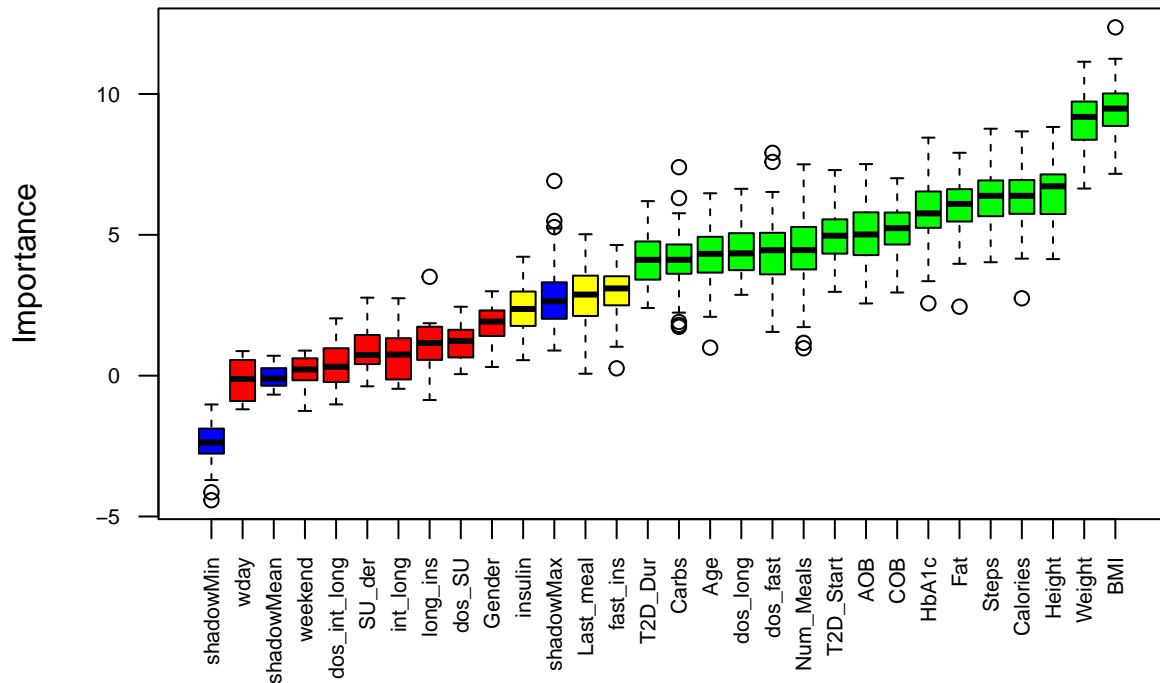
Boruta feature selection with lifestyle condition.

```
boruta_output_life <- Boruta(Night_class_hypo_out ~ ., data=full_data_life_plus_train, doTrace=0)
boruta_signif_life <- getSelectedAttributes(boruta_output_life, withTentative = TRUE)
roughFixMod <- TentativeRoughFix(boruta_output_life)
boruta_signif <- getSelectedAttributes(roughFixMod)
```

```
# This plot shows the distribution of selected, tentative and
# rejected variables for the Lifestyle condition
imps_life <- attStats(boruta_output_life)
imps_life = imps_life[imps_life$decision != 'Rejected', c('meanImp', 'decision')]
plot(boruta_output_life, cex.axis=.7, las=2, xlab="", main="Variable Importance")
```



## Variable Importance



## Feature Importance plot based on Boruta feature selection

This plot shows the mean Importance of variables over multiple iterations based on mean decrease accuracy as part of an RF approach.

This plot is used in the report to emphasize which variables performed best.

```
varimp_full <- imps_full[order(-imps_full$meanImp), ] %>% select(-decision)%>%
  rownames_to_column("feature") %>% rename(importance = meanImp) %>%
  mutate(Condition = "Full data")
varimp_life<- imps_life[order(-imps_life$meanImp), ] %>% select(-decision) %>%
  rownames_to_column("feature") %>% rename(importance = meanImp) %>%
  mutate(Condition = "Lifestyle data")

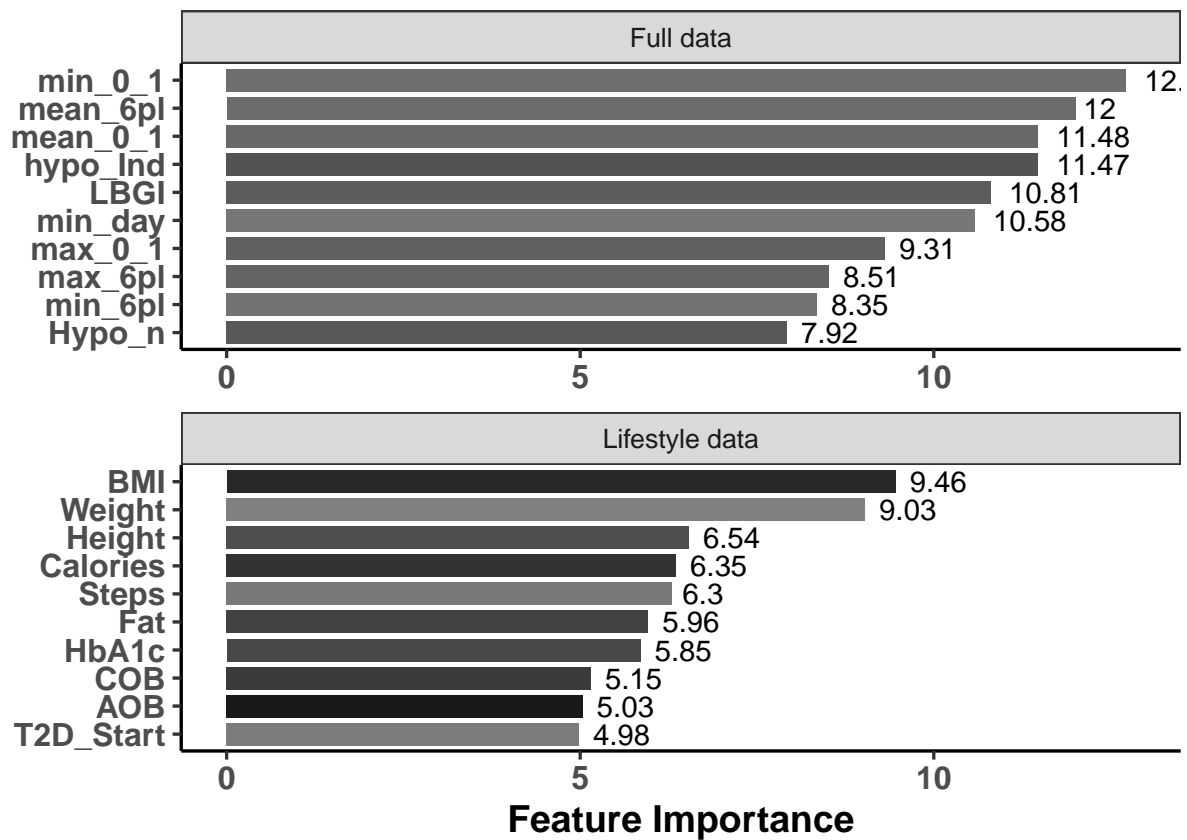
all_top_features <- rbind(varimp_full[1:10,],varimp_life[1:10,])

all_top <- ggplot(data = all_top_features,
  aes(x = reorder(feature, importance), y = importance, fill = feature)) +
  facet_wrap(~ Condition,scales="free",ncol=1) +
  geom_bar(stat="identity", width = 0.8) + scale_fill_grey(start = 0.1, end = 0.5) +
  geom_text(aes(label = round(importance, 2)), color="black",
    size=4, hjust = -0.25, position = position_dodge(width = 0.5),inherit.aes = TRUE) +
  theme_bw(base_size = 12) + theme(legend.position = "none") +
  scale_x_discrete(guide = guide_axis(n.dodge=1)) + theme_bw(base_size = 13) +
  theme(legend.position="",
```

```

panel.border = element_blank(), # remove panel borders
panel.grid.major = element_blank(), # remove major grid lines
panel.grid.minor = element_blank(), # remove minor grid lines
axis.line = element_line(), # set axis lines to bold
axis.text = element_text(size = 12, face = "bold"), # set axis text to bold
axis.title = element_text(size = 14, face = "bold"),
plot.background = element_blank(), # remove plot background
panel.background = element_blank() +
coord_flip() +
ylim(0, max(all_top_features$importance) * 1.01) + labs(y = "Feature Importance", x = "")
print(all_top)

```



For Supplementary Material also the 25 best variables based on MI overall

```

all_top_features <- mut_inf[1:25,]

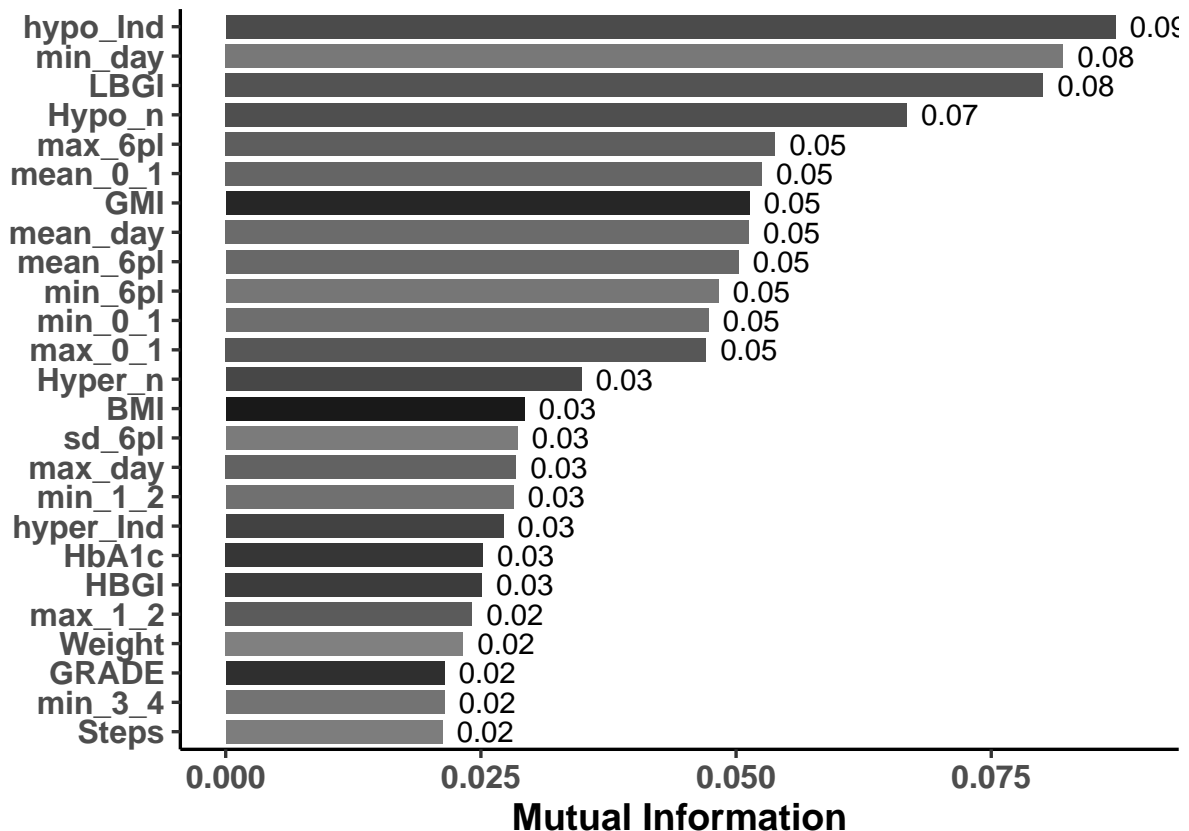
all_top <- ggplot(data = all_top_features,
  aes(x = reorder(feature, importance), y = importance, fill = feature)) +
  geom_bar(stat="identity", width = 0.8) + scale_fill_grey(start = 0.1, end = 0.5) +
  geom_text(aes(label = round(importance, 2)), color="black",
    size=4, hjust = -0.25, position = position_dodge(width = 0.5), inherit.aes = TRUE) +
  theme_bw(base_size = 12) + theme(legend.position = "none") +
  scale_x_discrete(guide = guide_axis(n.dodge=1)) + theme_bw(base_size = 13) +
  theme(legend.position=""),
  panel.border = element_blank(), # remove panel borders
  panel.grid.major = element_blank(), # remove major grid lines

```

```

panel.grid.minor = element_blank(), # remove minor grid lines
axis.line = element_line(), # set axis lines to bold
axis.text = element_text(size = 12, face = "bold"), # set axis text to bold
axis.title = element_text(size = 14, face = "bold"),
plot.background = element_blank(), # remove plot background
panel.background = element_blank() +
coord_flip() +
ylim(0, max(all_top_features$importance) * 1.02) + labs(y = "Mutual Information", x = "")
print(all_top)

```



## Extract common important features across approaches

The overlapping features are directly included and some variables are further included based on Expert knowledge. ### Full data Condition

```

# Find the common features selected by both methods
missing_bor <- left_join(varimp_full, mut_inf, by="feature")
missing_bor <- missing_bor[is.na(missing_bor$importance.y), ]

missing_mi <- left_join(mut_inf, varimp_full, by="feature")
missing_mi <- missing_mi[is.na(missing_mi$importance.y), ]

common_vars <- intersect(varimp_full$feature, mut_inf$feature)
common_vars

```

```
## [1] "min_0_1"      "mean_6pl"      "mean_0_1"      "hypo_Ind"      "LBGI"
## [6] "min_day"      "max_0_1"      "max_6pl"      "min_6pl"      "Hypo_n"
## [11] "GMI"          "mean_day"      "min_1_2"      "mean_2_3"      "max_day"
## [16] "hyper_Ind"    "min_2_3"      "HBGI"         "max_2_3"      "GRADE"
## [21] "tir_63_140"   "mean_1_2"      "max_1_2"      "Hyper_n"      "sd_6pl"
## [26] "AOB"          "mean_4_5"      "min_3_4"      "min_4_5"      "mean_5_6"
## [31] "mean_3_4"     "T2D_Start"     "max_3_4"
```

```
missing_bor$feature
```

```
## [1] "slope_day" "sd_day"     "slope_0_1" "tir_70_180" "max_5_6"
```

```
missing_mi$feature
```

```
## [1] "BMI"          "HbA1c"        "Weight"        "Steps"         "Height"        "T2D_Dur"
## [7] "slope_1_2"    "Age"          "max_4_5"       "Calories"      "min_5_6"
```

```
# Variables selected based on Expert knowledge to be included
```

```
extra_expert_full <- c("slope_0_1", "tir_70_180", "BMI", "HbA1c", "Weight", "Height", "T2D_Dur", "Age")
```

## Lifestyle Condition

Same for the Lifestyle condition. As the results of Boruta differ this has to be done again. The overlapping features are directly included and some variables are further included based on Expert knowledge.

```
# Find the common features selected by both methods
```

```
missing_bor_life <- left_join(varimp_life, mut_inf_life, by="feature")
missing_bor_life <- missing_bor_life[is.na(missing_bor_life$importance.y), ]
```

```
missing_mi_life <- left_join(mut_inf_life, varimp_life, by="feature")
missing_mi_life <- missing_mi_life[is.na(missing_mi_life$importance.y), ]
```

```
common_vars_life <- intersect(varimp_life$feature, mut_inf_life$feature)
common_vars_life
```

```
## [1] "BMI"          "Weight"        "Height"        "Calories"      "Steps"         "Fat"
## [7] "HbA1c"        "COB"          "AOB"          "T2D_Start"     "Num_Meals"     "dos_long"
## [13] "Age"          "Carbs"        "T2D_Dur"      "Last_meal"
```

```
missing_bor_life$feature
```

```
## [1] "dos_fast" "fast_ins" "insulin"
```

```
missing_mi_life$feature
```

```
## [1] "wday"
```

```
# Variables selected based on Expert knowledge to be included
extra_expert_life <- c("dos_fast","fast_ins")
```

## Final Selection both data conditions

Combing the data sets and seeing how many features are selected

```
# Combining selected and expert knowledge for both data conditions
```

```
full_data_sel <- full_data %>% select(Night_class_hypo_out,common_vars,extra_expert_full)
```

```
full_data_life_plus_sel <- full_data_life_plus %>%
  select(Night_class_hypo_out,common_vars_life,extra_expert_life)
colnames(full_data_sel) # -1 for outcome variable = Features
```

```
## [1] "Night_class_hypo_out" "min_0_1" "mean_6pl"
## [4] "mean_0_1" "hypo_Ind" "LBGI"
## [7] "min_day" "max_0_1" "max_6pl"
## [10] "min_6pl" "Hypo_n" "GMI"
## [13] "mean_day" "min_1_2" "mean_2_3"
## [16] "max_day" "hyper_Ind" "min_2_3"
## [19] "HBGI" "max_2_3" "GRADE"
## [22] "tir_63_140" "mean_1_2" "max_1_2"
## [25] "Hyper_n" "sd_6pl" "AOB"
## [28] "mean_4_5" "min_3_4" "min_4_5"
## [31] "mean_5_6" "mean_3_4" "T2D_Start"
## [34] "max_3_4" "slope_0_1" "tir_70_180"
## [37] "BMI" "HbA1c" "Weight"
## [40] "Height" "T2D_Dur" "Age"
```

```
colnames(full_data_life_plus_sel) # -1 for outcome variable = Features
```

```
## [1] "Night_class_hypo_out" "BMI" "Weight"
## [4] "Height" "Calories" "Steps"
## [7] "Fat" "HbA1c" "COB"
## [10] "AOB" "T2D_Start" "Num_Meals"
## [13] "dos_long" "Age" "Carbs"
## [16] "T2D_Dur" "Last_meal" "dos_fast"
## [19] "fast_ins"
```

## Saved data and feature selection process

Data is saved for further analyses.

```
write.csv(full_data_sel,"Data/full_data_sel.csv", row.names = FALSE)
write.csv(full_data_life_plus_sel,"Data/full_data_life_plus_sel.csv", row.names = FALSE)

feature_selection <- list(boruta_full= boruta_output_full, boruta_sig_full = boruta_signif_full,
  Vari_imp_full = varimp_full,boruta_life= boruta_output_life,
  boruta_sig_life = boruta_signif_life, Vari_imp_life = varimp_life)
saveRDS(feature_selection, file = "Data/feature_selection.RData")
```