Untitled

```
library(DALEXtra)
library(SummarizedExperiment)
library(glue)
library(parsnip)
library(tidymodels)
library(tidyverse)
library(vip)
theme_set(theme_bw())
```

We'll study the Type I Diabetes data. The two objects below consider the studies combined/separately.

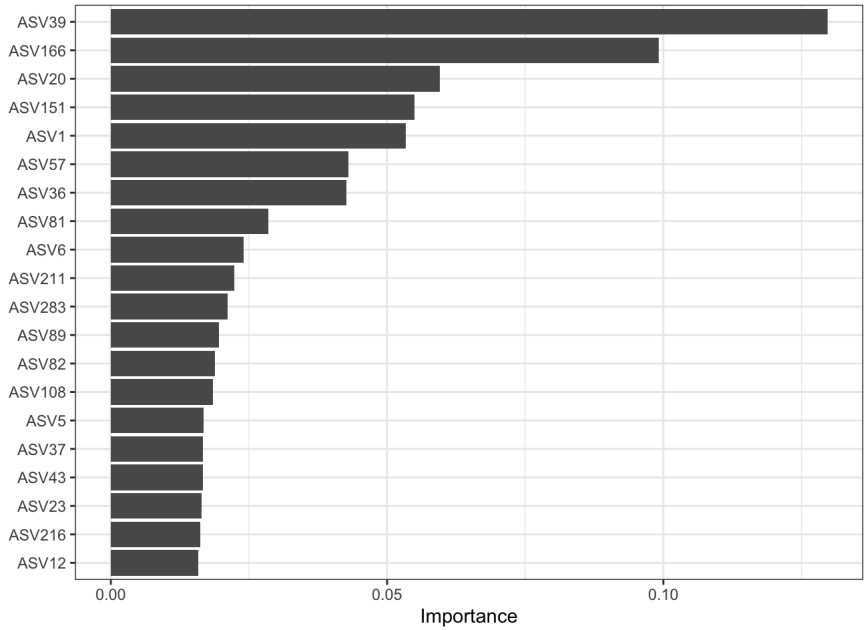
```
load("T1D.rda")
se <- se[, colData(se)$disease %in% c("healthy", "T1D")]
x <- t(assay(se)) |>
  as_tibble() %>%
  set_names(glue("ASV{seq_along(.)}"))

combined_data <- bind_cols(
  x,
  y = factor(colData(se)$disease),
  study_name = colData(se)$study_name
)

split_data <- combined_data %>%
  split(.$study_name)
```

We'll fit models in the two extremes: completely separate fits, and completely combined.

```
gbm <- boost_tree(mode = "classification", trees = 50)
combined_fit <- fit(gbm, y ~ ., data =
        select(combined_data, -study_name))
separate_fits <- map(split_data, ~ fit(gbm, y ~ ., data =
        select(., -study_name)))
```

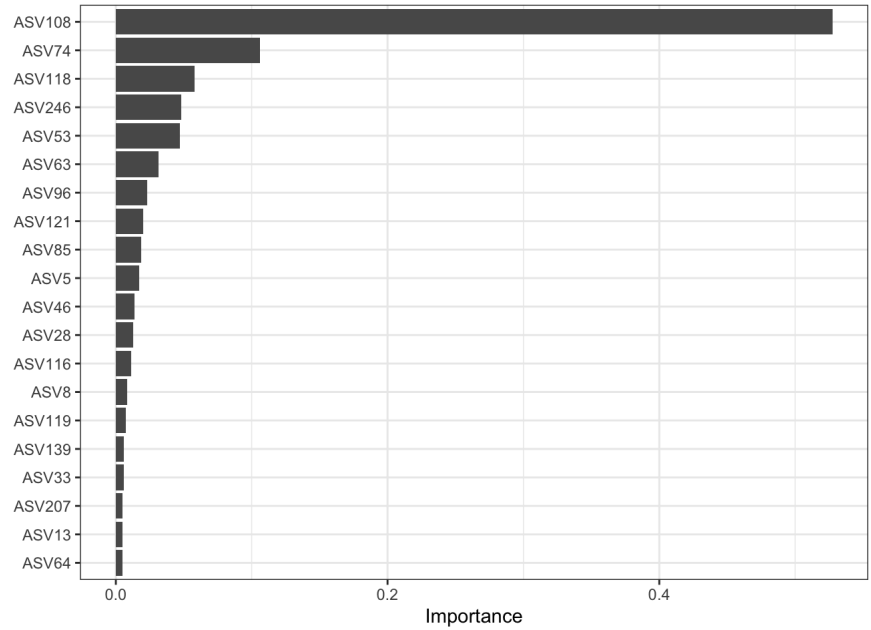
Let's compare the features that are considered important across models.

```
vip(combined_fit, num_features = 20)
```

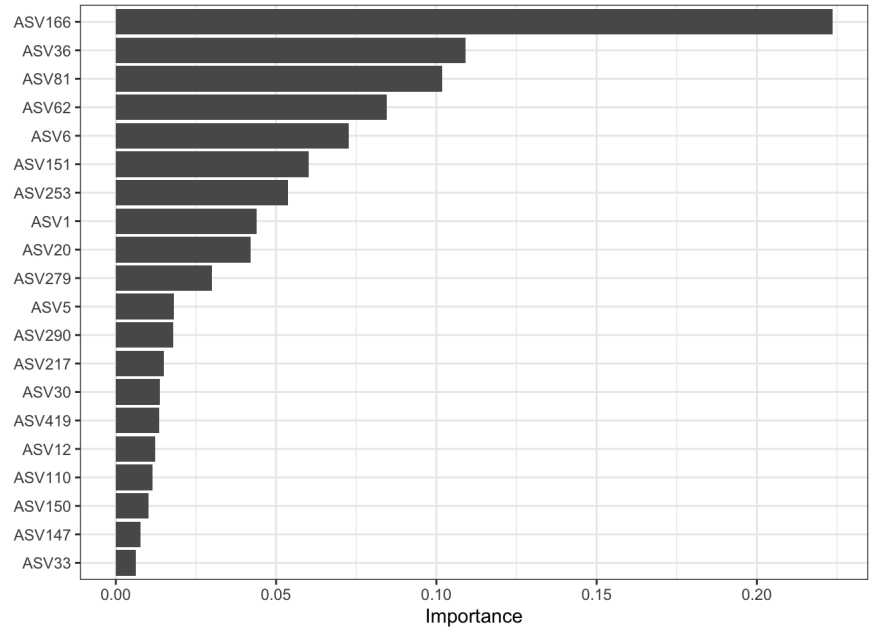



```
map(separate_fits, ~ vip(., num_features = 20))
```

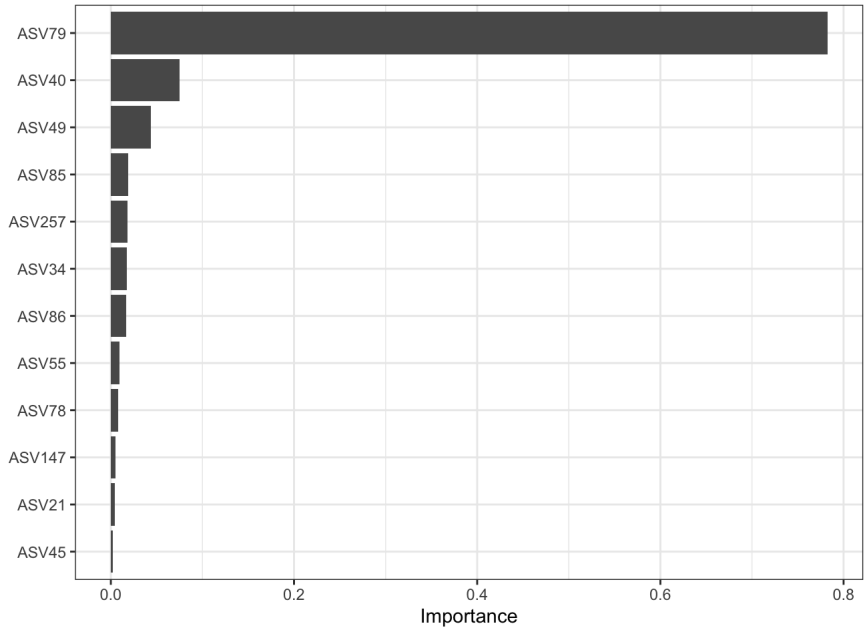
```
## $`Heitz-BuschartA_2016`
```




```
##
## $KosticAD_2015
```


ASV166
ASV36
ASV81
ASV62
ASV6
ASV151
ASV253
ASV1
ASV20
ASV279
ASV5
ASV290
ASV217
ASV30
ASV419
ASV12
ASV110
ASV150
ASV147
ASV33
0.00
0.05
0.10
0.15
0.20
Importance

```
##
## $LiJ_2014
```



We can interpret the results as well.

```
focus_taxa <- c("ASV39", "ASV166")
explainer <- explain_tidymodels(combined_fit, data =
        select(combined_data, -study_name:-y), y =
        combined_data$y)
```

```
## Preparation of a new explainer is initiated
##   -> model label       :  model_fit  (  default  )
##   -> data              :  206  rows  678  cols
##   -> data              :  tibble converted into a
data.frame
##   -> target variable   :  206  values
##   -> predict function  :  yhat.model_fit  will be used
(  default  )
##   -> predicted values  :  No value for predict function
target column. (  default  )
##   -> model_info        :  package parsnip , ver. 1.0.3
```

```
, task classification (  default  )
##   -> model_info        :  Model info detected
classification task but 'y' is a factor .  (  WARNING  )
##   -> model_info        :  By deafult classification
tasks supports only numercical 'y' parameter.
##   -> model_info        :  Consider changing to
numerical vector with 0 and 1 values.
##   -> model_info        :  Otherwise I will not be able
to calculate residuals or loss function.
##   -> predicted values  :  numerical, min =  0.002174497
, mean =  0.5824698 , max =  0.9986486
##   -> residual function :  residual_function
##   -> residuals         :  numerical, min =  0 , mean =
0 , max =  0
##   A new explainer has been created!
```

```
profiles <- model_profile(explainer, variables =
        focus_taxa)
plot(profiles, geom = "profiles", variables = focus_taxa)
```

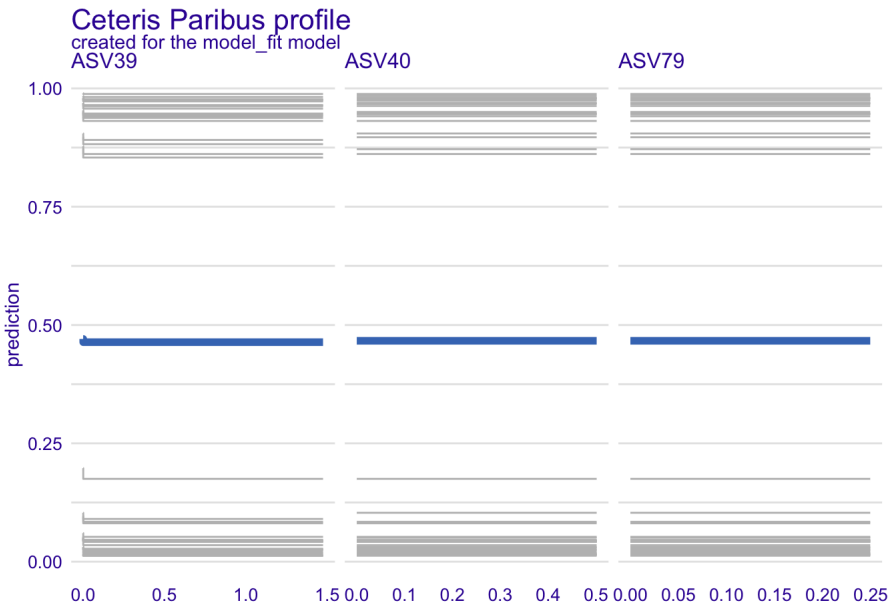
```
explainers <- separate_fits |>
  map2(split_data, ~ explain_tidymodels(.x, data =
         select(.y, -study_name:-y), y = .x$y))
```

```
## Preparation of a new explainer is initiated
##   -> model label       :  model_fit  (  default  )
##   -> data              :  45  rows  678  cols
##   -> data              :  tibble converted into a
data.frame
##   -> target variable   :  not specified! (  WARNING  )
##   -> predict function  :  yhat.model_fit  will be used
(  default  )
##   -> predicted values  :  No value for predict function
target column. (  default  )
##   -> model_info        :  package parsnip , ver. 1.0.3
, task classification (  default  )
##   -> model_info        :  Model info detected
classification task but 'y' is a NULL .  (  WARNING  )
##   -> model_info        :  By deafult classification
tasks supports only numercical 'y' parameter.
##   -> model_info        :  Consider changing to
numerical vector with 0 and 1 values.
##   -> model_info        :  Otherwise I will not be able
to calculate residuals or loss function.
##   -> predicted values  :  numerical, min =  0.01252091
, mean =  0.4666678 , max =  0.9882011
##   -> residual function :  residual_function
##   A new explainer has been created!
## Preparation of a new explainer is initiated
##   -> model label       :  model_fit  (  default  )
##   -> data              :  120  rows  678  cols
##   -> data              :  tibble converted into a
data.frame
##   -> target variable   :  not specified! (  WARNING  )
##   -> predict function  :  yhat.model_fit  will be used
(  default  )
##   -> predicted values  :  No value for predict function
```

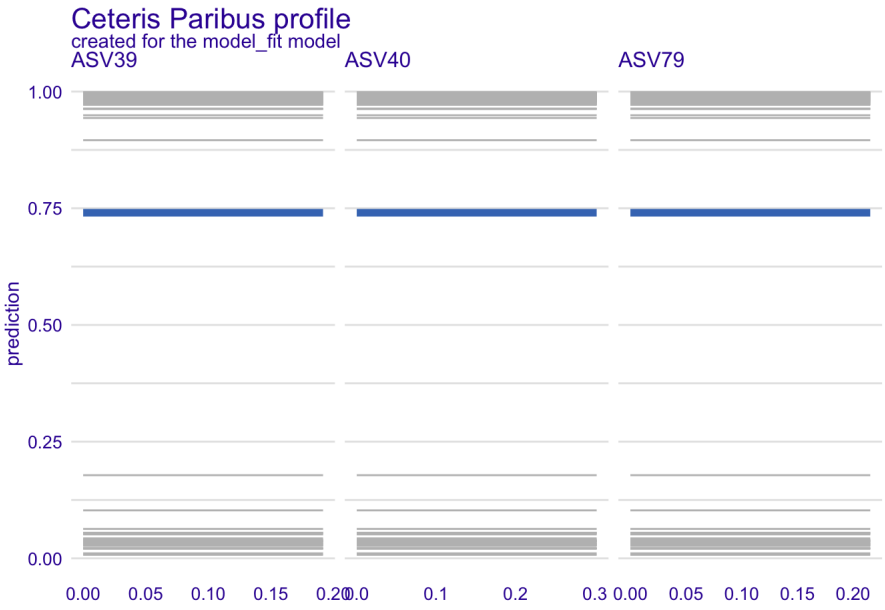
```
, task classification (  default  )
##   -> model_info        :  Model info detected
classification task but 'y' is a NULL .  (  WARNING  )
##   -> model_info        :  By deafult classification
tasks supports only numercical 'y' parameter.
##   -> model_info        :  Consider changing to
numerical vector with 0 and 1 values.
##   -> model_info        :  Otherwise I will not be able
to calculate residuals or loss function.
##   -> predicted values  :  numerical, min =  0.007546365
, mean =  0.7415414 , max =  0.9986223
##   -> residual function :  residual_function
##   A new explainer has been created!
## Preparation of a new explainer is initiated
##   -> model label       :  model_fit  (  default  )
##   -> data              :  41  rows  678  cols
##   -> data              :  tibble converted into a
data.frame
##   -> target variable   :  not specified! (  WARNING  )
##   -> predict function  :  yhat.model_fit  will be used
(  default  )
##   -> predicted values  :  No value for predict function
target column. (  default  )
##   -> model_info        :  package parsnip , ver. 1.0.3
, task classification (  default  )
##   -> model_info        :  Model info detected
classification task but 'y' is a NULL .  (  WARNING  )
##   -> model_info        :  By deafult classification
tasks supports only numercical 'y' parameter.
##   -> model_info        :  Consider changing to
numerical vector with 0 and 1 values.
##   -> model_info        :  Otherwise I will not be able
to calculate residuals or loss function.
##   -> predicted values  :  numerical, min =  0.01206541
, mean =  0.243906 , max =  0.9622628
##   -> residual function :  residual_function
##   A new explainer has been created!


focus_taxa <- c("ASV79", "ASV40", "ASV39")
explainers |>
```

```
  map(~ model_profile(., variables = focus_taxa)) |>
  map(~ plot(., geom = "profiles", variables =
         focus_taxa))
```

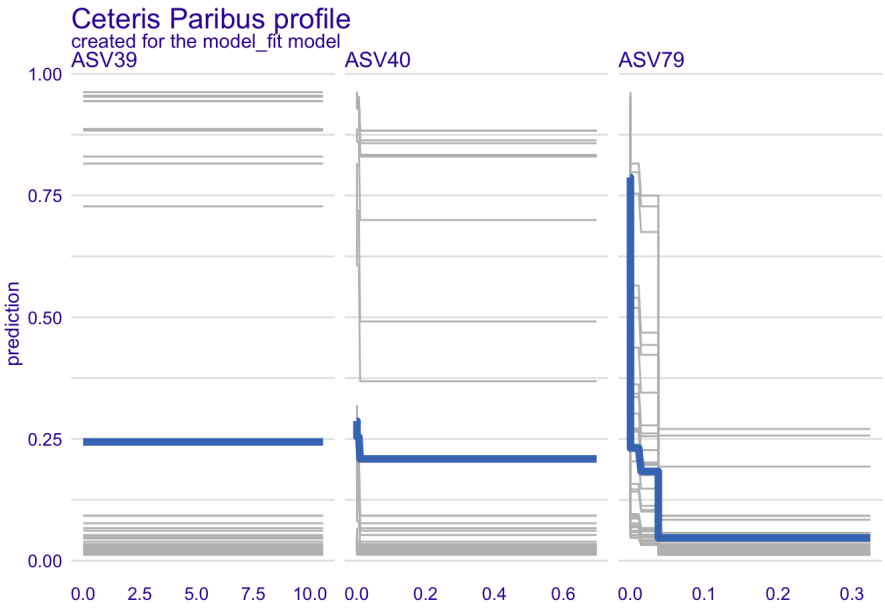
```
## $`Heitz-BuschartA_2016`
```



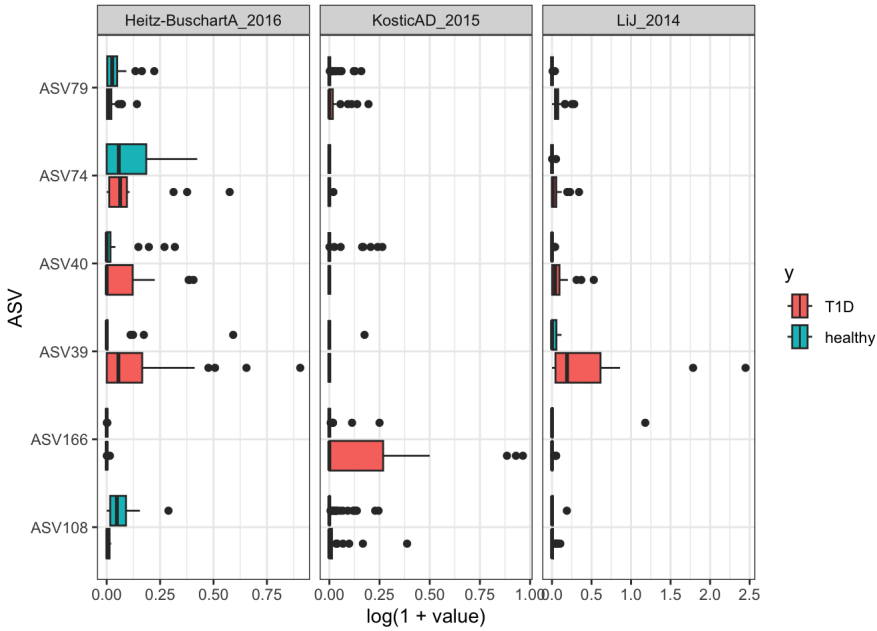

```
##
## $KosticAD_2015
```


Ceteris Paribus profile
created for the model_fit model
ASV39
ASV40
ASV79
1.00
0.75
0.50
0.25
0.00
prediction
0.00
0.05
0.10
0.15
0.20
0.0
0.1
0.2
0.3

```
##
## $LiJ_2014
```




```
focus_taxa <- c("ASV39", "ASV166", "ASV40", "ASV79",
         "ASV108", "ASV74")
combined_long <- combined_data |>
  select(y, study_name, focus_taxa) |>
  pivot_longer(starts_with("ASV"), names_to = "ASV")

ggplot(combined_long) +
  geom_boxplot(aes(log(1 + value), ASV, fill = y)) +
  facet_grid(. ~ study_name, scales = "free")
```


Heitz-BuschartA_2016
KosticAD_2015
LiJ_2014
ASV
ASV79
ASV74
ASV40
ASV39
ASV166
ASV108
0.00
0.25
0.50
0.75
1.00
0.0
0.5
1.0
1.5
2.0
2.5
log(1 + value)
y
T1D
healthy