

# Aspect importance - demo

*Katarzyna Pękala*

*2019-10-29*

## Demo

### Aspect importance

Aspect importance function provides instance-level explanations for the groups of explanatory variables. It enables grouping predictors into entities called aspects. Afterwards, it can calculate the contribution of those aspects to the prediction.

To illustrate how the function works, we use titanic example. We build random forest model, group features into aspects and choose new observation to be explained. Then we build DALEX explainer and use it to call aspect importance function. Finally, we print and plot function results. We can observe that **personal** (age and gender) variables have the biggest contribution to the prediction. This contribution is of a positive type.

```
library("DALEX")
library("randomForest")
library(ingredients)
titanic <- titanic_imputed
titanic$country <- NULL
titanic_without_target <- titanic[,colnames(titanic)!="survived"]

aspects_titanic <-
  list(
    wealth = c("class", "fare"),
    family = c("sibsp", "parch"),
    personal = c("age", "gender"),
    embarked = "embarked"
  )

passenger <- titanic_without_target[4,]

model_titanic_rf <- randomForest(factor(survived) == "yes" ~ gender + age +
                                class + embarked + fare + sibsp + parch,
                                data = titanic)

predict(model_titanic_rf, passenger)

#>           4
#> 0.6183772

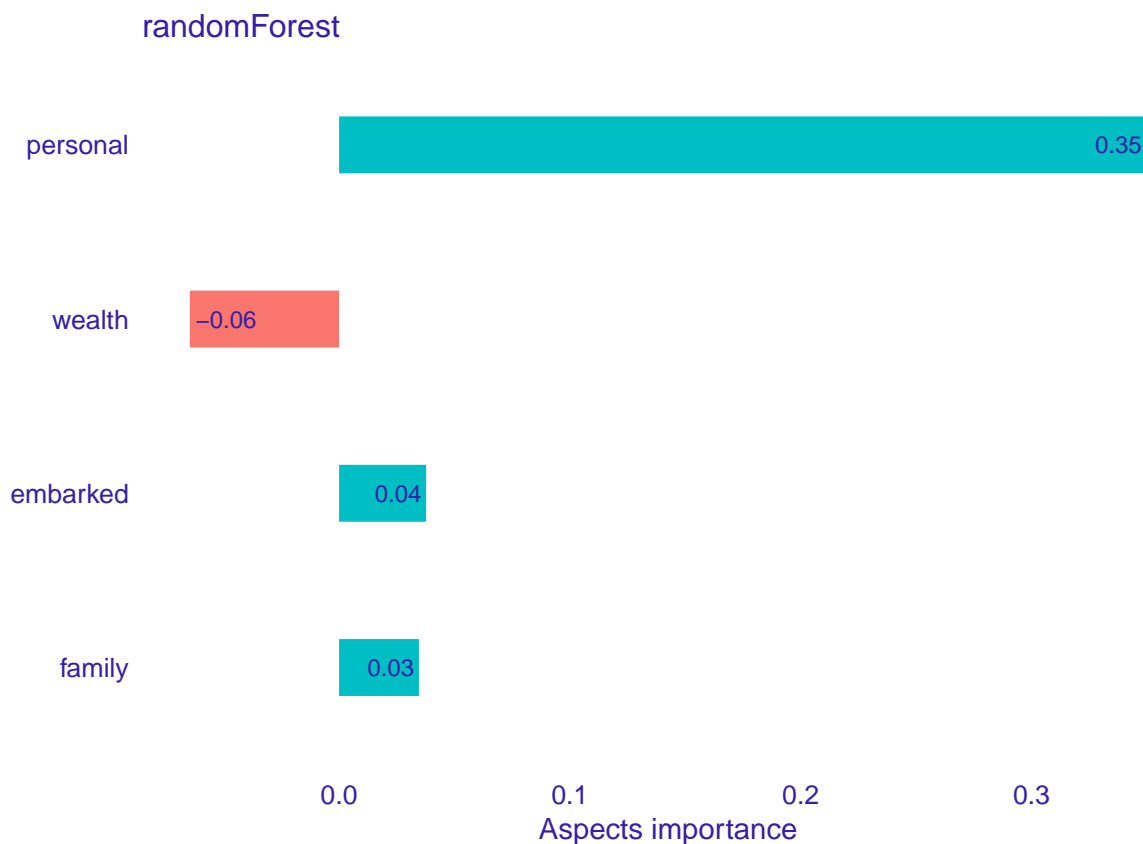
explain_titanic_rf <- explain(model_titanic_rf,
                             data = titanic_without_target,
                             y = titanic$survived == "yes",
                             predict_function = predict,
                             verbose = FALSE)
```

```
titanic_rf_ai <- aspect_importance(x = explain_titanic_rf,
                                   new_observation = passenger,
                                   aspects = aspects_titanic)

titanic_rf_ai
```

```
#>   aspects importance   features
#> 4 personal    0.34982 age, gender
#> 2  wealth   -0.06466 class, fare
#> 5 embarked    0.03770   embarked
#> 3  family     0.03457 sibsp, parch
```

```
plot(titanic_rf_ai, add_importance = TRUE)
```



## Automated grouping

In examples described above, we had to manually group features into aspects. Aspect importance provides `group_variables()` - function that automatically groups features for us, based on the features correlation. Function only works on numeric variables.

Below, we test aspect importance function on another dataset. But this time we build aspect list by running `run_group_variables()` (with cut off level set on 0.6). As a result, we get a list of variables groups (aspects) where absolute value of features' pairwise correlation is at least at 0.6.

```

library(DALEX)
data("apartments")
apartments_num <- apartments[,unlist(lapply(apartments, is.numeric))]
apartments_no_target <- apartments_num[,-1]
new_observation_apartments <- apartments_num[10,]
model_apartments <- lm(m2.price ~ ., data = apartments_num)
aspects_apartments <- group_variables(apartments_no_target, 0.6)
predict(model_apartments, new_observation_apartments)

```

```

#>      10
#> 2875.794

```

```

explain_apartments <- explain(model_apartments,
                              data = apartments_no_target,
                              y = apartments$m2.price,
                              predict_function = predict,
                              verbose = FALSE)
apartments_ai <- aspect_importance(x = explain_apartments,
                                   new_observation = new_observation_apartments,
                                   aspects = aspects_apartments,
                                   N = 1000, show_cor = TRUE)

apartments_ai

```

```

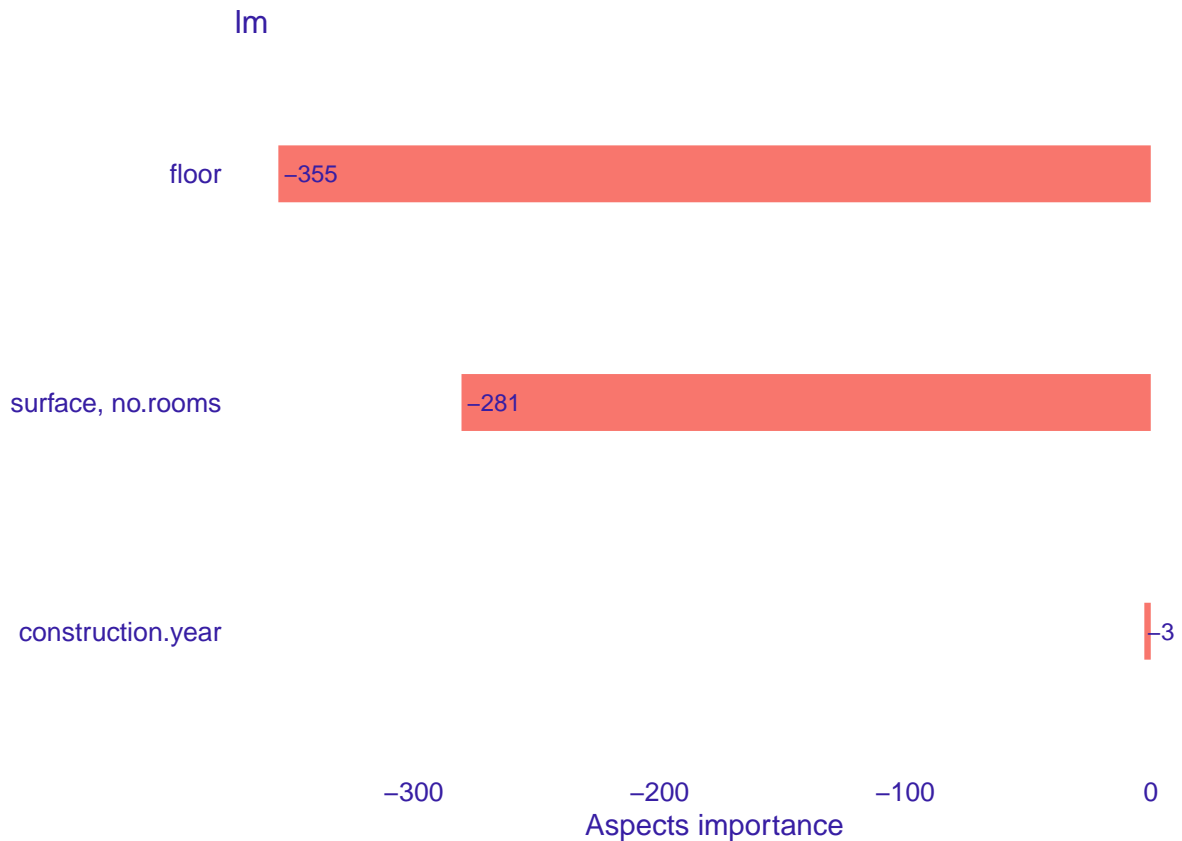
#>      aspects importance      features min_cor sign
#> 4 aspect.group3    -355.08         floor      NA
#> 3 aspect.group2   -280.53 surface, no.rooms 0.9174349 pos
#> 2 aspect.group1    -2.62 construction.year      NA

```

```

plot(apartments_ai, aspects_on_axis = FALSE, add_importance = TRUE,
      digits_to_round = 0)

```



## Triplot

**Triplot** is one more tool that allows us to better understand the inner workings of a black box model. After calling `triplot` and providing it with model, dataset and observation to be explained, it illustrates (in one place):

- single aspect importance - aspect importance results when every aspect has only one feature,
- hierarchical aspects importance (description can be found in vignette),
- order of grouping features into aspects in `group_variables()`.

```
triplot(model_apartments, apartments_no_target,
        new_observation = new_observation_apartments, N = 5000,
        add_importance_labels = FALSE, axis_lab_size = 8, abbrev_labels = 11)
```

