Local Exploration, Explanation and Visualisation of Predictive Models

Introduction

When decisions from Machine Learning models are confronted with humans, following questions sparkle: Why? Why this decision was made? Which features support this decision? Can we trust this decision? How would it change if a single feature is slightly different?

Methods for local exploration/explanation are designed to improve out understanding of model predictions that refer to a particular observation.

Use-Case

As an use-case we are using **HR** dataset from the **DALEX** package. Five variables are used for a classification problem, would a given employee shall be fired, promoted or left as it is. It's an artificial dataset designed in a way that variable age and gender are in interaction.

```
library("DALEX")
head(HR, 2)
##   gender      age    hours evaluation salary   status
## 1   male 32.58267 41.88626          3      1    fired
## 2 female 41.21104 36.34339          2      5    fired
```

Presented tools are part of DALEXverse, set of tools for model agnostic exploration, explanation and visualisation of predictive models.

Find mode information about DALEX at the website https://pbiecek.github.io/DALEX_docs/ or online book https://pbiecek.github.io/PM_VEE/

Model preparation

Model exploration starts with a model to explore, and the observation of interest.

1. First we need to train a model

```
library("randomForest")
rf_model <- randomForest(
   status ~ gender + age + hours +
   evaluation + salary, data = HR)
```

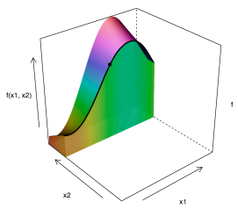
2. Then we need to build an explainer - model enriched with additional metadata like validation data, predict function, true labels. Use the **explain()** function from the **DALEX** package.

```
library("DALEX")
explainer_rf <- explain(rf_model,
   data = HR,
   y = HR$status == "fired")
```

3. Local explainers work for a selected observation / point in a feature space.

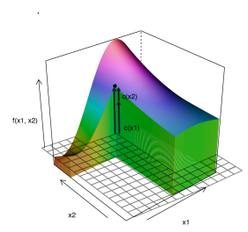
```
John1960 <- data.frame(
   gender = factor("male",
       levels = c("male","female")),
   age = 57.7,
   hours = 42.3,
   evaluation = 2,
   salary = 2)
```

Calculation of local explainers



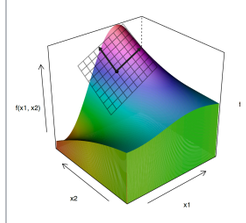
What-If scenarios with Ceteris Paribus Profiles are supported with the **ceterisParibus** package. unction. Use the function **ceteris_paribus()** with an explainer and the observation of interest as first arguments. You may also select variables of interest.

```
library("ceterisParibus")
cp_rf <- ceteris_paribus(explainer_rf, John1960,
    variables = c("age", "hours"))
cp_rf
## Top profiles    :
##     gender      age hours evaluation salary
## 1     male 20.00389  42.3          2      2
## 1.1   male 20.35994  42.3          2      2
##     _yhat_ _vname_ _ids_       _label_
## 1   0.4234617     age     1 randomForest
## 1.1 0.3761229     age     1 randomForest
```



Variable attributions are supported with the **breakDown** package.

```
library("breakDown")
bd_rf <- break_down(explainer_rf, John1960)
bd_rf
##                             contribution
## (Intercept)                        0.364
## * hours = 42                       0.161
## * age:gender = 58:male             0.120
## * salary = 2                      -0.045
##  final_prognosis                   0.648
```



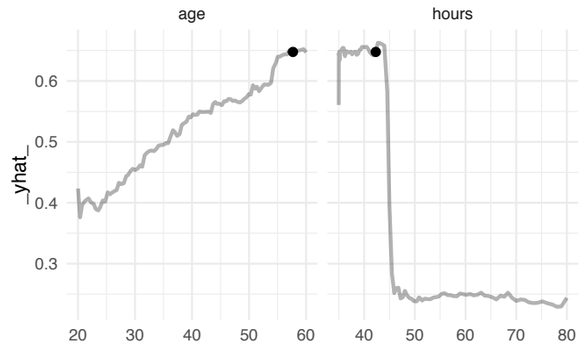
Local model approximation is supported with the **live** package.

```
library("live")
lm_rf <- local_approximation(explainer_rf,
   John1960, target_variable_name = "status")
lm_rf
## Explanation model:
## Name:  regr.lm
## R-squared: 0.9889
```

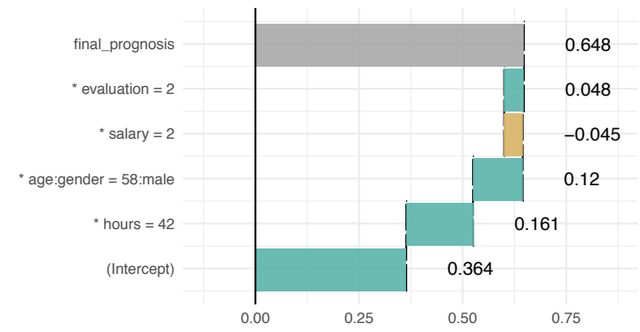
Visualisation of explainers

The generic function **plot()** works for every local explainer.

```
plot(cp_rf)
```




```
plot(bd_rf)
```




```
plot(lm_rf)
```

Variable		N	Estimate	p
gender	male	449		Reference
	female	51	−0.26 (−0.26, −0.26)	<0.001
age		500	0.00 (−0.01, 0.01)	0.65
hours		500	0.01 (0.00, 0.01)	0.02
evaluation		500	0.01 (0.00, 0.01)	<0.001
salary		500	−0.01 (−0.01, −0.01)	<0.001
(Intercept)			0.30 (−0.20, 0.80)	0.24

−0.20 0.20.40.6