# xgboost

```
library(mlr3)
library(mlr3learners)
library(mlr3filters)
library(tidyverse)
library(ggplot2)
```

## xgboost Upload Rate Prediction

```
data_dir = "../datasets/"
dataset_ul = read_csv(str_c(data_dir, "dataset_ul.csv"), col_types = cols(scenario=col_factor()))
dataset_ul_prediction = dataset_ul %>% select(
  # TODO: encode scenario as described here:
  # https://mlr3gallery.mlr-org.com/posts/2020-01-31-encode-factors-for-xgboost/
  #scenario,
  velocity_mps,
  acceleration_mpss,
  rsrp_dbm,
  rsrq_db,
  rssnr_db,
  cqi,
  ta,
  payload_mb,
  f_mhz,
  throughput_mbits
)

# remove missing values
dataset_ul_prediction = dataset_ul_prediction %>% drop_na()
glimpse(dataset_ul_prediction)
```

```
## Rows: 6,168
## Columns: 10
## $ velocity_mps      <dbl> 11.80, 11.83, 11.70, 11.45, 11.49, 7.93, 8.15, 8....
## $ acceleration_mpss <dbl> 0.13, 0.03, 0.06, -0.32, -0.26, 0.23, 0.24, 0.32,...
## $ rsrp_dbm          <dbl> -99, -85, -121, -84, -97, -96, -74, -108, -111, -...
## $ rsrq_db           <dbl> -9, -5, -15, -6, -12, -12, -5, -9, -13, -11, -6, ...
## $ rssnr_db          <dbl> -1, 22, -8, 11, -2, 5, 29, 2, 6, 11, 13, 16, -3, ...
## $ cqi               <dbl> 8, 10, 4, 13, 9, 5, 15, 2, 6, 15, 12, 9, 6, 11, 1...
## $ ta                <dbl> 9, 7, 63, 4, 7, 7, 4, 21, 16, 7, 4, 4, 7, 16, 4, ...
## $ payload_mb        <dbl> 1.0, 4.0, 6.0, 2.0, 6.0, 5.0, 4.0, 10.0, 0.1, 7.0...
## $ f_mhz             <dbl> 1750, 1720, 1770, 1720, 1750, 1750, 1720, 1770, 1...
## $ throughput_mbits  <dbl> 4.66, 24.52, 1.29, 14.86, 3.97, 6.52, 16.27, 3.18...
```

```
task = TaskRegr$new(
  id = "ul_prediction",
  backend = dataset_ul_prediction,
```

```
  target = "throughput_mbits"
)
task
```

```
## <TaskRegr:ul_prediction> (6168 x 10)
## * Target: throughput_mbits
## * Properties: -
## * Features (9):
##   - dbl (9): acceleration_mpss, cqi, f_mhz, payload_mb, rsrp_dbm,
##     rsrq_db, rssnr_db, ta, velocity_mps
```

```
learner_xgboost = lrn("regr.xgboost", nrounds=10)
```

```
resampling = rsmp("holdout", ratio = 0.8)
result = resample(
  task = task,
  learner = learner_xgboost,
  resampling = resampling
)
```

```
## INFO  [15:39:48.047] Applying learner 'regr.xgboost' on task 'ul_prediction' (iter 1/1)
## [15:39:48] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecate
```

```
result
```

```
## <ResampleResult> of 1 iterations
## * Task: ul_prediction
## * Learner: regr.xgboost
## * Warnings: 0 in 0 iterations
## * Errors: 0 in 0 iterations
```

```
# get r^2
result$aggregate(msr("regr.rsq"))
```

```
##  regr.rsq
## 0.8116962
```

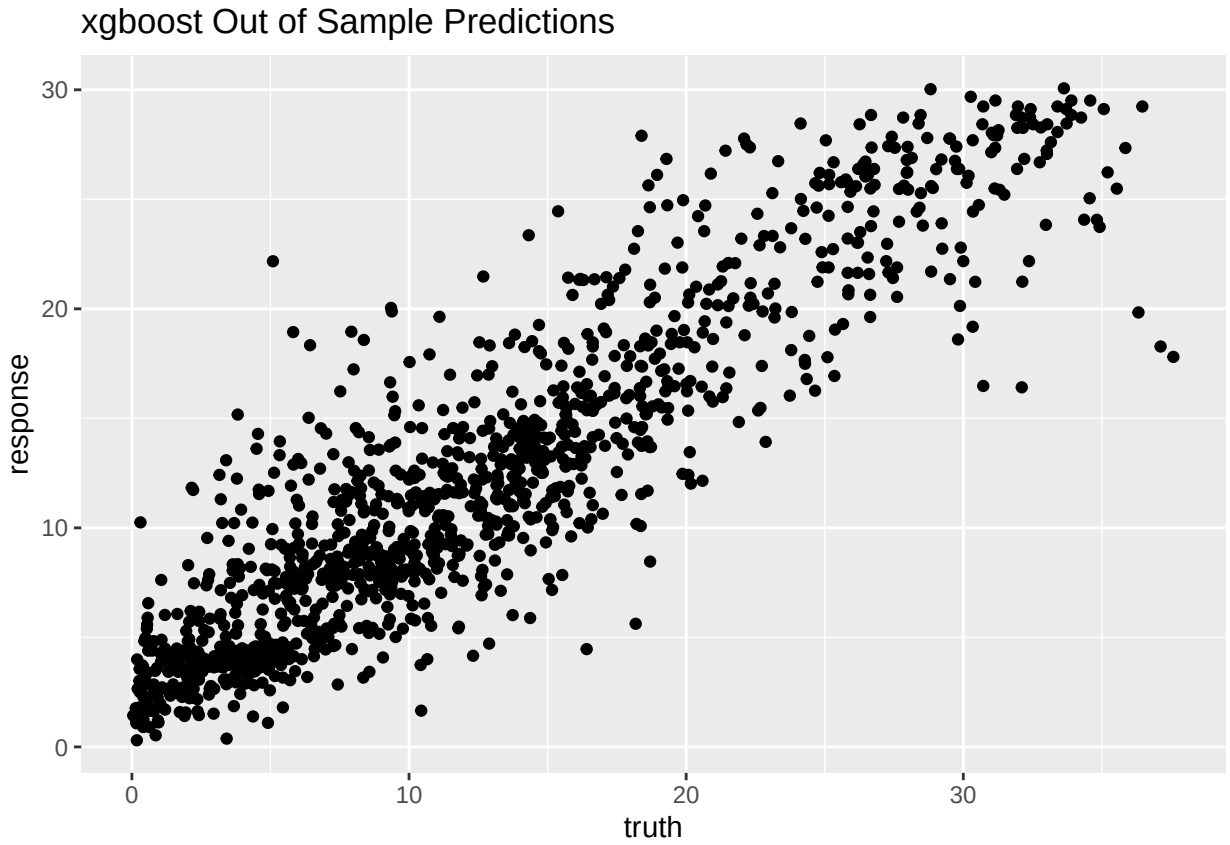
```
# get MSE
result$aggregate(msr("regr.mse"))
```

```
## regr.mse
## 14.42459
```

```
# get MAE
result$aggregate(msr("regr.mae"))
```

```
## regr.mae
## 2.789909
```

```
predictions = as.data.table(result$prediction())
ggplot(predictions) +
  geom_point(aes(x=truth, y=response)) +
  ggtitle("xgboost Out of Sample Predictions")
```


xgboost Out of Sample Predictions
response
truth
0
10
20
30