

# mlr3 Exercises Day 1 - Ames Housing Dataset

This exercise is intended to use mlr3 to perform a benchmark analysis on the *Ames Dataset*.

The main objectives of this exercise are as follows:

- To build machine learning models able to predict house price based on house features
- To analyze and compare models performance in order to choose the best model

## Accessing the dataset

The dataset is available on Kaggle <https://bit.ly/2l0uWoz>. Kaggle is a platform which provides data science competitions and datasets which can be used to get familiar with typical machine learning methods.

## Importing the data

```
train_set = read.csv("data/ames_housing_train.csv")
```

1. Load the mlr3 and mlr3learners packages.

```
library(mlr3)
library(mlr3learners)
```

2. Create a regression task object.

```
task = TaskRegr$new(id = "ames_housing", backend = train_set, target = "SalePrice")
task
```

```
## <TaskRegr:ames_housing> (1953 x 81)
## * Target: SalePrice
## * Properties: -
## * Features (80):
##   - fct (43): Alley, Bldg.Type, Bsmt.Cond, Bsmt.Exposure,
##     Bsmt.Qual, BsmtFin.Type.1, BsmtFin.Type.2, Central.Air,
##     Condition.1, Condition.2, Electrical, Exter.Cond, Exter.Qual,
##     Exterior.1st, Exterior.2nd, Fence, Fireplace.Qu, Foundation,
##     Functional, Garage.Cond, Garage.Finish, Garage.Qual,
##     Garage.Type, Heating, Heating.QC, House.Style, Kitchen.Qual,
##     Land.Contour, Land.Slope, Lot.Config, Lot.Shape, MS.Zoning,
##     Mas.Vnr.Type, Misc.Feature, Neighborhood, Paved.Drive,
##     Pool.QC, Roof.Matl, Roof.Style, Sale.Condition, Sale.Type,
##     Street, Utilities
##   - int (37): Bedroom.AbvGr, Bsmt.Full.Bath, Bsmt.Half.Bath,
##     Bsmt.Unf.SF, BsmtFin.SF.1, BsmtFin.SF.2, Enclosed.Porch,
##     Fireplaces, Full.Bath, Garage.Area, Garage.Cars,
##     Garage.Yr.Blt, Gr.Liv.Area, Half.Bath, Kitchen.AbvGr,
##     Lot.Area, Lot.Frontage, Low.Qual.Fin.SF, MS.SubClass,
##     Mas.Vnr.Area, Misc.Val, Mo.Sold, Open.Porch.SF, Overall.Cond,
##     Overall.Qual, PID, Pool.Area, Screen.Porch, TotRms.AbvGrd,
##     Total.Bsmt.SF, Wood.Deck.SF, X1st.Flr.SF, X2nd.Flr.SF,
##     X3Ssn.Porch, Year.Built, Year.Remod.Add, Yr.Sold
```

3. Create a list of learning algorithms which you want to use in the benchmark.

```
# get a featureless learner as baseline
# Additionally, we can train a regression tree, a knn learner
```

```
# for different values of k and a random forest (ranger)
learners = list(
  featureless = lrn("regr.featureless"),
  knn3 = lrn("regr.kknn", id = "regr.knn3", k = 3),
  knn7 = lrn("regr.kknn", id = "regr.knn7", k = 7), #default
  knn15 = lrn("regr.kknn", id = "regr.knn15", k = 15),
  knn30 = lrn("regr.kknn", id = "regr.knn30", k = 30),
  tree = lrn("regr.rpart"),
  random_forest = lrn("regr.ranger")
)
```

4. Create a resampling object for your benchmark evaluation.

```
# compare via 10-fold cross validation
resamplings = rsmp("cv", folds = 10)
```

5. Create a grid corresponding to the planned benchmark including the task, all learners and the resampling strategy.

```
# create a BenchmarkDesign object
design = benchmark_grid(task, learners, resamplings)
print(design)
```

```
##           task                learner    resampling
## 1: <TaskRegr> <LearnerRegrFeatureless> <ResamplingCV>
## 2: <TaskRegr>          <LearnerRegrKKNN> <ResamplingCV>
## 3: <TaskRegr>          <LearnerRegrKKNN> <ResamplingCV>
## 4: <TaskRegr>          <LearnerRegrKKNN> <ResamplingCV>
## 5: <TaskRegr>          <LearnerRegrKKNN> <ResamplingCV>
## 6: <TaskRegr>          <LearnerRegrRpart> <ResamplingCV>
## 7: <TaskRegr>          <LearnerRegrRanger> <ResamplingCV>
```

6. Run the benchmark

```
# execute the benchmark
bmr = benchmark(design)
```

```
## INFO [20:21:41.288] Benchmark with 70 resampling iterations
## INFO [20:21:41.289] Applying learner 'regr.featureless' on task 'ames_housing' (iter 1/10)
## INFO [20:21:41.296] Applying learner 'regr.featureless' on task 'ames_housing' (iter 2/10)
## INFO [20:21:41.321] Applying learner 'regr.featureless' on task 'ames_housing' (iter 3/10)
## INFO [20:21:41.334] Applying learner 'regr.featureless' on task 'ames_housing' (iter 4/10)
## INFO [20:21:41.343] Applying learner 'regr.featureless' on task 'ames_housing' (iter 5/10)
## INFO [20:21:41.353] Applying learner 'regr.featureless' on task 'ames_housing' (iter 6/10)
## INFO [20:21:41.360] Applying learner 'regr.featureless' on task 'ames_housing' (iter 7/10)
## INFO [20:21:41.367] Applying learner 'regr.featureless' on task 'ames_housing' (iter 8/10)
## INFO [20:21:41.373] Applying learner 'regr.featureless' on task 'ames_housing' (iter 9/10)
## INFO [20:21:41.378] Applying learner 'regr.featureless' on task 'ames_housing' (iter 10/10)
## INFO [20:21:41.384] Applying learner 'regr.knn3' on task 'ames_housing' (iter 1/10)
## INFO [20:21:41.528] Applying learner 'regr.knn3' on task 'ames_housing' (iter 2/10)
## INFO [20:21:41.680] Applying learner 'regr.knn3' on task 'ames_housing' (iter 3/10)
## INFO [20:21:41.787] Applying learner 'regr.knn3' on task 'ames_housing' (iter 4/10)
## INFO [20:21:41.908] Applying learner 'regr.knn3' on task 'ames_housing' (iter 5/10)
## INFO [20:21:42.010] Applying learner 'regr.knn3' on task 'ames_housing' (iter 6/10)
## INFO [20:21:42.385] Applying learner 'regr.knn3' on task 'ames_housing' (iter 7/10)
## INFO [20:21:42.484] Applying learner 'regr.knn3' on task 'ames_housing' (iter 8/10)
## INFO [20:21:42.579] Applying learner 'regr.knn3' on task 'ames_housing' (iter 9/10)
```

```

## INFO [20:21:42.684] Applying learner 'regr.knn3' on task 'ames_housing' (iter 10/10)
## INFO [20:21:42.779] Applying learner 'regr.knn7' on task 'ames_housing' (iter 1/10)
## INFO [20:21:43.122] Applying learner 'regr.knn7' on task 'ames_housing' (iter 2/10)
## INFO [20:21:43.219] Applying learner 'regr.knn7' on task 'ames_housing' (iter 3/10)
## INFO [20:21:43.319] Applying learner 'regr.knn7' on task 'ames_housing' (iter 4/10)
## INFO [20:21:43.425] Applying learner 'regr.knn7' on task 'ames_housing' (iter 5/10)
## INFO [20:21:43.524] Applying learner 'regr.knn7' on task 'ames_housing' (iter 6/10)
## INFO [20:21:43.632] Applying learner 'regr.knn7' on task 'ames_housing' (iter 7/10)
## INFO [20:21:43.965] Applying learner 'regr.knn7' on task 'ames_housing' (iter 8/10)
## INFO [20:21:44.065] Applying learner 'regr.knn7' on task 'ames_housing' (iter 9/10)
## INFO [20:21:44.204] Applying learner 'regr.knn7' on task 'ames_housing' (iter 10/10)
## INFO [20:21:44.323] Applying learner 'regr.knn15' on task 'ames_housing' (iter 1/10)
## INFO [20:21:44.430] Applying learner 'regr.knn15' on task 'ames_housing' (iter 2/10)
## INFO [20:21:44.534] Applying learner 'regr.knn15' on task 'ames_housing' (iter 3/10)
## INFO [20:21:44.646] Applying learner 'regr.knn15' on task 'ames_housing' (iter 4/10)
## INFO [20:21:44.992] Applying learner 'regr.knn15' on task 'ames_housing' (iter 5/10)
## INFO [20:21:45.099] Applying learner 'regr.knn15' on task 'ames_housing' (iter 6/10)
## INFO [20:21:45.208] Applying learner 'regr.knn15' on task 'ames_housing' (iter 7/10)
## INFO [20:21:45.311] Applying learner 'regr.knn15' on task 'ames_housing' (iter 8/10)
## INFO [20:21:45.415] Applying learner 'regr.knn15' on task 'ames_housing' (iter 9/10)
## INFO [20:21:45.525] Applying learner 'regr.knn15' on task 'ames_housing' (iter 10/10)
## INFO [20:21:45.623] Applying learner 'regr.knn30' on task 'ames_housing' (iter 1/10)
## INFO [20:21:45.739] Applying learner 'regr.knn30' on task 'ames_housing' (iter 2/10)
## INFO [20:21:46.095] Applying learner 'regr.knn30' on task 'ames_housing' (iter 3/10)
## INFO [20:21:46.205] Applying learner 'regr.knn30' on task 'ames_housing' (iter 4/10)
## INFO [20:21:46.316] Applying learner 'regr.knn30' on task 'ames_housing' (iter 5/10)
## INFO [20:21:46.436] Applying learner 'regr.knn30' on task 'ames_housing' (iter 6/10)
## INFO [20:21:46.560] Applying learner 'regr.knn30' on task 'ames_housing' (iter 7/10)
## INFO [20:21:46.674] Applying learner 'regr.knn30' on task 'ames_housing' (iter 8/10)
## INFO [20:21:46.799] Applying learner 'regr.knn30' on task 'ames_housing' (iter 9/10)
## INFO [20:21:46.923] Applying learner 'regr.knn30' on task 'ames_housing' (iter 10/10)
## INFO [20:21:47.297] Applying learner 'regr.rpart' on task 'ames_housing' (iter 1/10)
## INFO [20:21:47.333] Applying learner 'regr.rpart' on task 'ames_housing' (iter 2/10)
## INFO [20:21:47.386] Applying learner 'regr.rpart' on task 'ames_housing' (iter 3/10)
## INFO [20:21:47.441] Applying learner 'regr.rpart' on task 'ames_housing' (iter 4/10)
## INFO [20:21:47.482] Applying learner 'regr.rpart' on task 'ames_housing' (iter 5/10)
## INFO [20:21:47.550] Applying learner 'regr.rpart' on task 'ames_housing' (iter 6/10)
## INFO [20:21:47.585] Applying learner 'regr.rpart' on task 'ames_housing' (iter 7/10)
## INFO [20:21:47.622] Applying learner 'regr.rpart' on task 'ames_housing' (iter 8/10)
## INFO [20:21:47.681] Applying learner 'regr.rpart' on task 'ames_housing' (iter 9/10)
## INFO [20:21:47.736] Applying learner 'regr.rpart' on task 'ames_housing' (iter 10/10)
## INFO [20:21:47.772] Applying learner 'regr.ranger' on task 'ames_housing' (iter 1/10)
## INFO [20:21:48.439] Applying learner 'regr.ranger' on task 'ames_housing' (iter 2/10)
## INFO [20:21:49.086] Applying learner 'regr.ranger' on task 'ames_housing' (iter 3/10)
## INFO [20:21:49.688] Applying learner 'regr.ranger' on task 'ames_housing' (iter 4/10)
## INFO [20:21:50.342] Applying learner 'regr.ranger' on task 'ames_housing' (iter 5/10)
## INFO [20:21:51.012] Applying learner 'regr.ranger' on task 'ames_housing' (iter 6/10)
## INFO [20:21:51.589] Applying learner 'regr.ranger' on task 'ames_housing' (iter 7/10)
## INFO [20:21:52.168] Applying learner 'regr.ranger' on task 'ames_housing' (iter 8/10)
## INFO [20:21:52.787] Applying learner 'regr.ranger' on task 'ames_housing' (iter 9/10)
## INFO [20:21:53.369] Applying learner 'regr.ranger' on task 'ames_housing' (iter 10/10)
## INFO [20:21:54.014] Finished benchmark

```

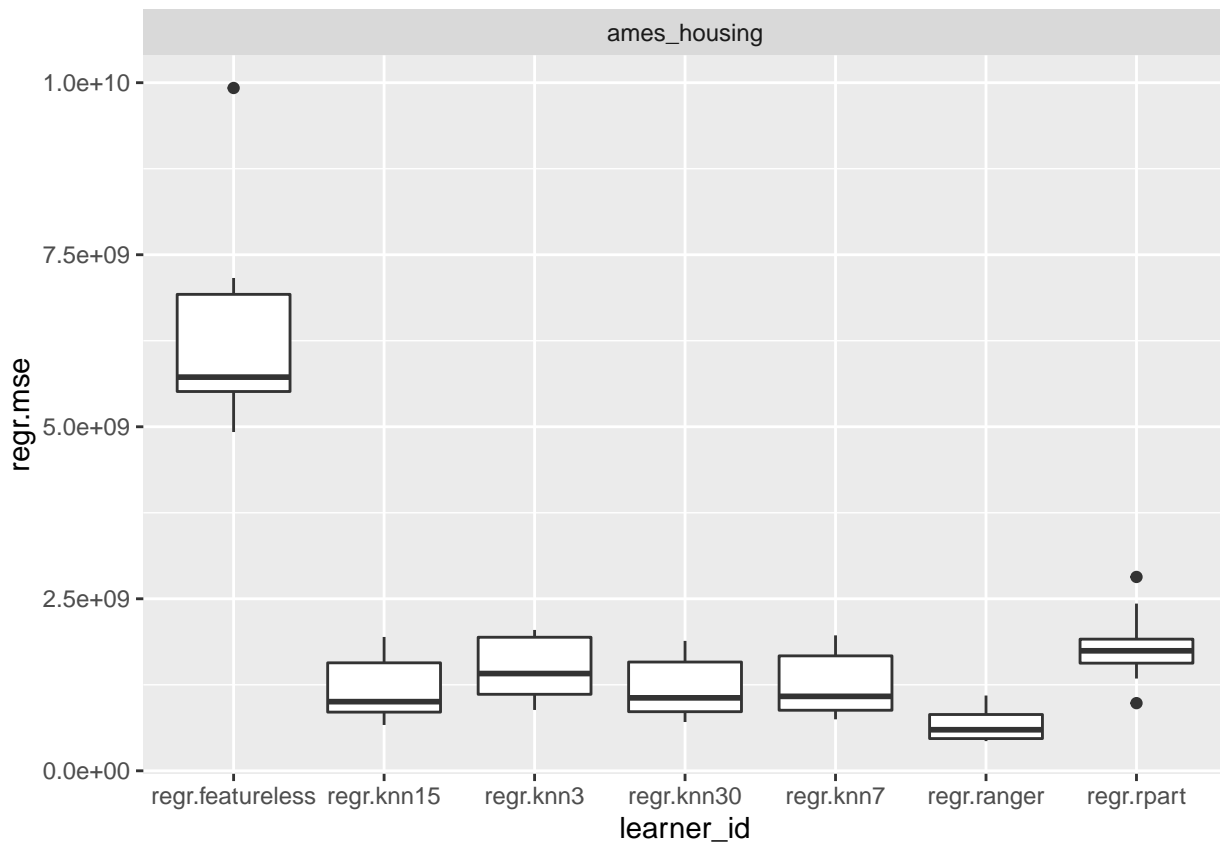
7. Use appropriate regression measures to measure the performance of each learner in the benchmark.

```
# get some measures: Mean Squared Error (which we use for the competition)
# and Mean Absolute Error
measures = mlr_measures$get(c("regr.mse", "regr.mae"))
bmr$aggregate(measures)
```

```
##      nr resample_result      task_id      learner_id resampling_id iters
## 1:  1 <ResampleResult> ames_housing regr.featureless          cv      10
## 2:  2 <ResampleResult> ames_housing      regr.knn3          cv      10
## 3:  3 <ResampleResult> ames_housing      regr.knn7          cv      10
## 4:  4 <ResampleResult> ames_housing      regr.knn15         cv      10
## 5:  5 <ResampleResult> ames_housing      regr.knn30         cv      10
## 6:  6 <ResampleResult> ames_housing      regr.rpart         cv      10
## 7:  7 <ResampleResult> ames_housing      regr.ranger         cv      10
##      regr.mse regr.mae
## 1: 6326910508  58307
## 2: 1492164747  22776
## 3: 1246064818  20849
## 4: 1180536914  20388
## 5: 1197349967  20706
## 6: 1804279680  28068
## 7:  673451134  15599
```

8. Use an appropriate plot to illustrate the benchmark results. Have e.g. a look at the `mlr3viz` package.

```
# create a nice boxplot
library(mlr3viz)
autoplot(bmr)
```



9. Finally, we choose the ranger as final algorithm, train it on the complete training data and predict on the test data.

```
test_set = read.csv("data/ames_housing_test.csv")
final_learner = learners$random_forest
final_learner$train(task)
pred = final_learner$predict_newdata(task, test_set)

# we can save the predictions as data.table and export them for Kaggle
pred = as.data.table(pred)
pred$truth = NULL
write.csv(pred, "data/ames_housing_submission_day1.csv")
```