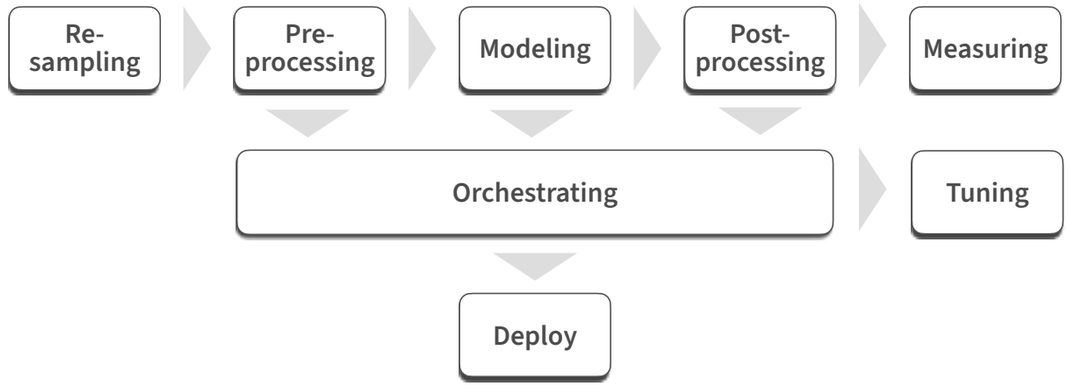
Machine learning with tidymodels : : **CHEAT SHEET**

Intro

tidymodels is a collection of R packages for modeling and machine learning that share a common design and grammar. Unlike most cheat sheets, which cover the functions of a single package, ***this cheatsheet maps the packages themselves***, grouping each by where it fits in the machine learning workflow.

Machine Learning workflow



Resampling

Split and resample data for honest evaluation

rsample - Create and manage resampling sets such as cross-validation, bootstraps, and validation splits.

spatialsample - Resample spatial data while respecting geographic structure.

Pre-processing

Prepare data for modeling

recipes - The preprocessing framework for building pipeable feature-engineering steps.

embed - Recipe steps that encode categorical predictors via target, likelihood, and entity embeddings.

textrecipes - Recipe steps that turn text into model-ready features.

themis - Recipe steps to rebalance class-imbalanced data with up-sampling, down-sampling, and SMOTE.

filtro - Filter-based supervised feature selection.

Modeling

Define and fit models through one consistent interface

parsnip - The core tidy interface every model plugs into, across many engines.

SPECIALIZED PROBLEMS

poissonreg - Poisson and count-data regression.

censored - Survival models for time-to-event outcomes.

multilevelmod - Mixed-effects and hierarchical models.

tidyclust - Clustering models under a tidy interface

tabby - For tabular deep-learning models, runs with brulee and tabpfn

CLASSIFICATION & REGRESSION

bonsai - Tree-based engines such as LightGBM and partykit.

baguette - Bagged ensembles of trees and MARS.

rules - Rule-based models such as Cubist, C5.0, and RuleFit.

discrim - Discriminant analysis and naive Bayes models.

plsmod - Partial least squares and other projection models.

agua - Interface to h2o models and AutoML.

Post-processing

Adjust predictions

probably - Tune classification thresholds and handle equivocal zones.

tailor - Post-process predictions through calibration and other sequential adjustments.

Measuring

Measure model quality

yardstick - Measure model performance with tidy metrics.

tidyposterior - Compare models across resamples using Bayesian methods.

Orchestrating

Tie the pieces together

workflows - Bundle preprocessing, model, and post-processing into one object.

workflowsets - Create and evaluate many workflows at once.

stacks - Build stacked ensembles from tuned models.

Machine learning with tidymodels : : **CHEAT SHEET**

Tuning

Optimize hyperparameters

tune - Run grid and iterative hyperparameter search.

dials - Define tuning parameters and build grids. It is often called for you by tune.

METHODS & OPTIMIZATION

finetune - Add search strategies such as racing and simulated annealing.

important - Measure predictor importance.

Deploy

Put models into production

PREPARE & SERVE

vetiver - Version, deploy, and monitor models in production.

butcher - Strip fitted models down to reduce object size.

applicable - Flag new samples that fall outside the training distribution.

RUN IN A DATABASE

tidypredict - Generate SQL to score models inside a database.

modeldb - Fit models directly in a database.

orbital - Convert workflows into portable equations that can run in-database.

Data

Datasets used in documentation, tests, and teaching

modeldata - Over 40 example datasets bundled for modeling.

modeldatatoo - Over half a dozen larger datasets downloaded on demand.

Deep Learning

R packages that implement or wrap tabular deep learning models

brulee - Torch-based models, from MLPs to ResNet and SAINT.

tabpfn - A pretrained transformer that predicts tabular data with no training.

Other

GENERAL

broom - Convert model objects into tidy tibbles.

infer - Run statistical inference and hypothesis tests.

corrr - Explore correlations in a data frame.

DEVELOPMENT

hardhat - Scaffold new modeling packages.

The **tidymodels** package installs, and loads a set of packages that are considered important during day-to-day machine learning development. It loads the following packages from tidymodels:

```
rsample       dials
recipes       workflows
parsnip       workflowsets
yardstick     broom
tailor        infer
tune          modeldata
```

It loads the following packages from the tidyverse:

```
dplyr         purrr
ggplot2       tidyr
```

Example

```
library(tidymodels)

# Split data and make CV folds
set.seed(857)

splits <- ames |>
  initial_split(prop = 0.8)

train <- training(splits)

folds <- train |>
  vfold_cv(v = 5)

# Feature engineering in a recipe
rec <- recipe(
  Sale_Price ~ Gr_Liv_Area +
    Year_Built + Bldg_Type,
  data = train
  ) |>
  step_log(Gr_Liv_Area, base = 10) |>
  step_dummy(
    all_nominal_predictors()
  ) |>
  step_normalize(
    all_numeric_predictors()
  )
```

Example (cont')

```
# Model with parameters to tune
mod <- decision_tree(
  cost_complexity = tune(),
  tree_depth = tune()
) |>
  set_engine("rpart") |>
  set_mode("regression")

# Bundle into a workflow
wf <- workflow() |>
  add_recipe(rec) |>
  add_model(mod)

# Tune over the folds
res <- wf |>
  tune_grid(
   resamples = folds,
   grid = 10
  )

# Finalize best, refit, test once
best <- res |>
  select_best(
   metric = "rmse"
  )

final <- wf |>
  finalize_workflow(best) |>
  last_fit(splits)

collect_metrics(final)

# Deploy the fitted workflow
library(vetiver)
library(pins)

v <- final |>
  extract_workflow() |>
  vetiver_model("ames_tree")

board <- board_temp()

vetiver_pin_write(board, v)
```