Response surface methodology

STAT 6160 • Experimental Design • Fall 2026

Siddharth Vishwanath

University of Virginia

§1. Hyperparameter tuning

A random forest for prediction

Which settings give the most accurate predictions?

The data have ten baseline measurements on each of 442 people with diabetes. Predict a disease progression score one year later.

One experimental run evaluates one set of forest hyperparameters.

Data from scikit-learn's load_diabetes. Lujan-Moreno et al. (2018) used RSM to tune a random forest for income classification.

The forest and its hyperparameters

A random forest fits trees to bootstrap samples. Each split considers a random subset of features. The forest averages the trees' predictions.

Setting	What it controls
`max_features`	How many features a split considers
`min_samples_leaf`	Minimum number of observations in a leaf
`max_depth`	How many splits a path through a tree can make

Use 96 trees at every setting.

The objective

Measure root mean squared error (RMSE) using one out-of-fold prediction $\hat{y}_{i,-f(i)}$ for each of the 331 tuning observations.

$$\text{Minimize} \;\; \Upsilon(\boldsymbol{\theta}) = \sqrt{\frac{1}{331}\sum_{i=1}^{331}\left(y_i - \hat{y}_{i,-f(i)}\right)^2}.$$

- $\boldsymbol{\theta}$ is the vector of three forest settings.
- Use the same five folds for every comparison.
- Set aside the remaining 111 observations for a final test.

A few settings to try

Features per split	Min. leaf size	Max. depth	CV RMSE
1	16	2	63.67
5	16	5	56.21
9	30	5	57.54

Like an oven temperature, depth is a setting we choose before the run. Use the middle row as the center of the experiment.

From the table, can we tell what increasing depth alone would do?

Response surface methodology

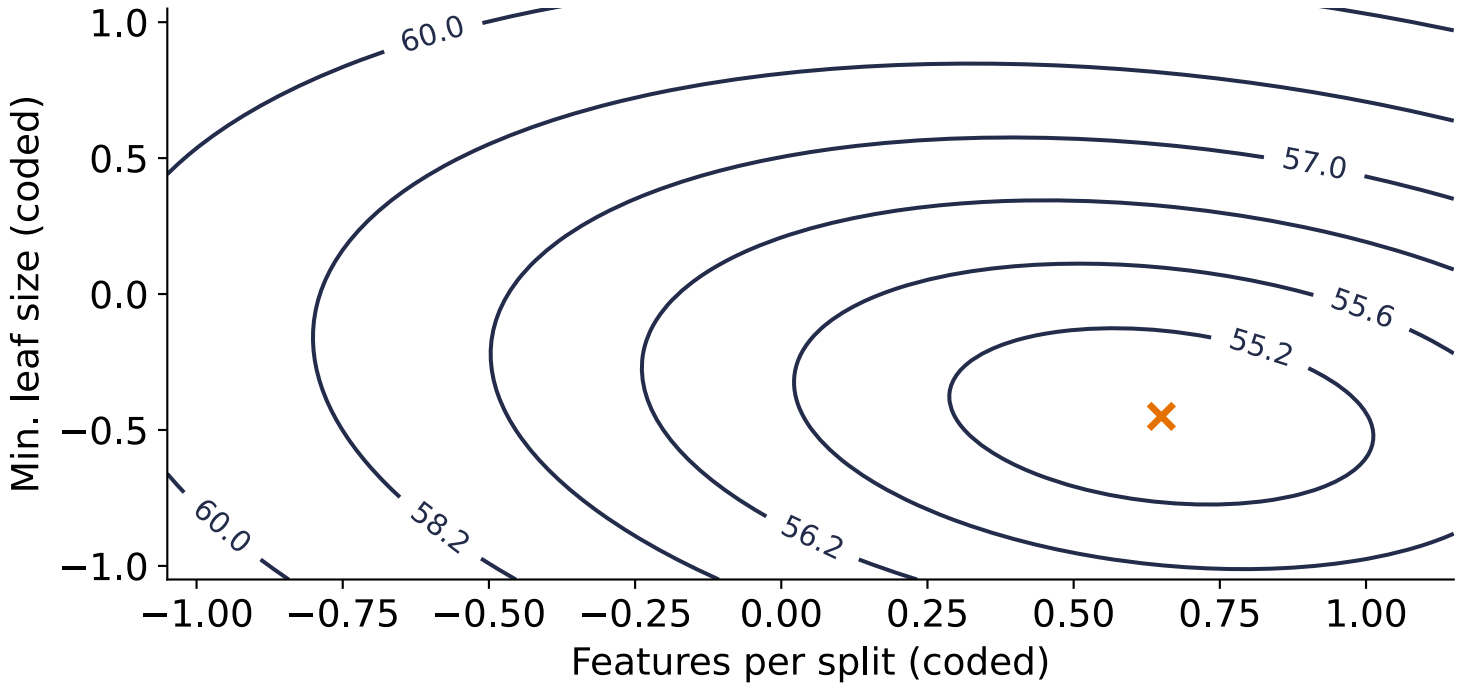
The **response surface** is mean performance as a function of the settings.

1. Run an experiment in a small region of settings.
2. Fit a regression to the measured performance.
3. Use the fitted surface to choose where to run the next experiment.

Look for lower RMSE in the random forest example.

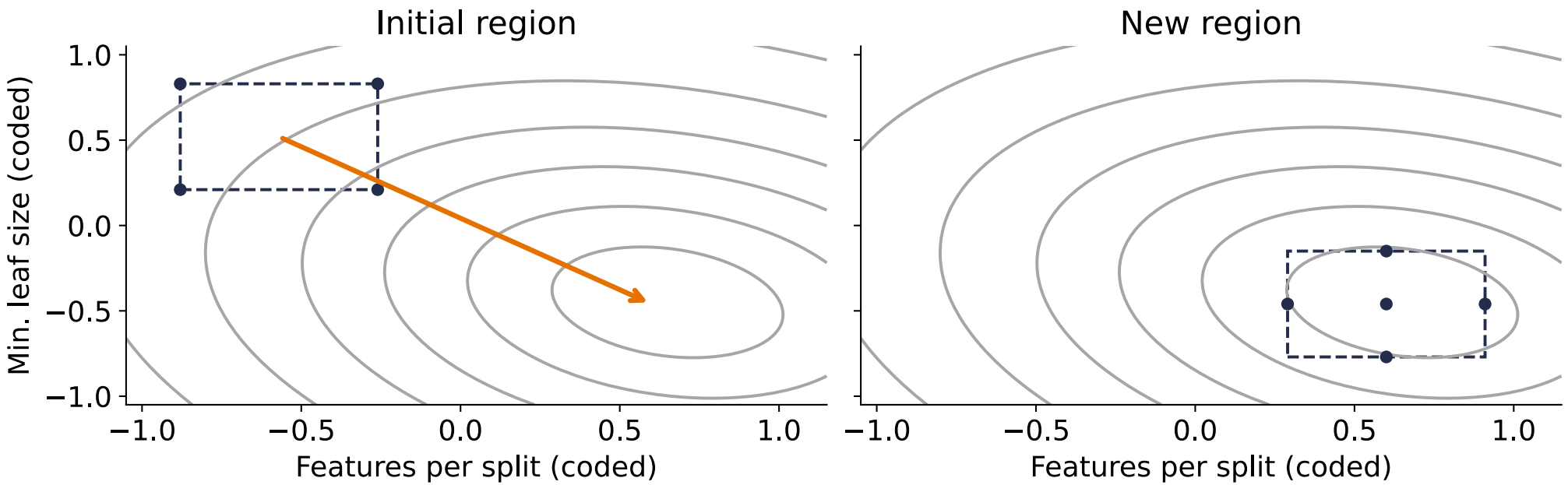
A slice through the surface

Each contour joins settings with the same error. Hold tree depth fixed to look at two settings at a time.



Illustrative surface. Contour labels are RMSE.

Two experimental regions



Far from a minimum, a plane can give a direction. Near a minimum, a quadratic can show how the error bends.

Illustrative contours. Wu and Hamada, §§10.2–10.4.

§2. Designs for curvature

Two factors at two levels

Vary features per split and minimum leaf size at -1 and $+1$. Hold depth fixed.

$$m(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2$$

- The $x_1 x_2$ term lets the effect of changing features depend on leaf size.
- At all four corners, $x_1^2 = x_2^2 = 1$.

Which column would either square duplicate?

The middle level

Code the middle of each factor range as 0. For features per split, $(1, 5, 9)$ becomes $(-1, 0, +1)$.

Run	(x_1, x_2)	(x_1^2, x_2^2)
A corner	$(1, 1)$	$(1, 1)$
The center	$(0, 0)$	$(0, 0)$
Change only features	$(1, 0)$	$(1, 0)$

What can the center tell us? What does the last row add?

The quadratic model

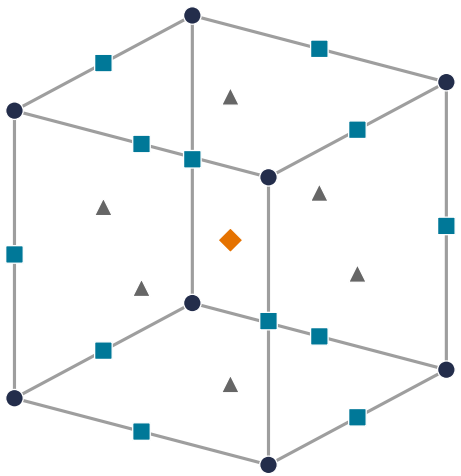
$$m(\boldsymbol{x}) = \beta_0 + \sum_j \beta_j x_j + \sum_{i<j} \beta_{ij} x_i x_j + \sum_j \beta_{jj} x_j^2$$

With three factors, there are $1 + 3 + 3 + 3 = 10$ coefficients.

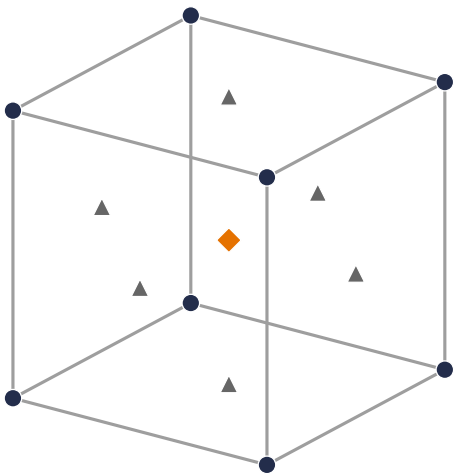
A 3^k factorial runs all combinations of the three levels. That takes 27 settings for three factors and 243 for five.

Three designs on the cube

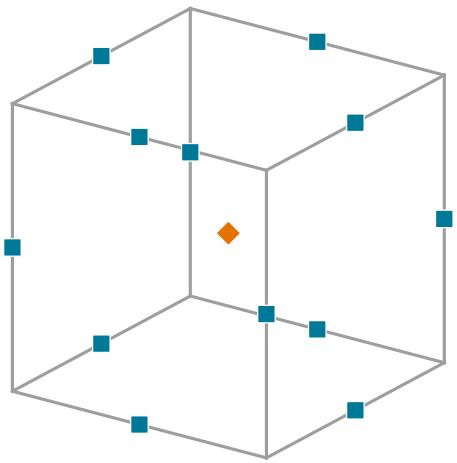
3^3 factorial



Central composite



Box–Behnken



Corners Edge midpoints Face centers Center

One point per distinct setting. The central composite design shown is face-centered.

Central composite and Box–Behnken

Design	Settings away from the center	Total runs
3^3 factorial	All 26 noncenter settings	29
Central composite	8 corners and 6 axial points	17
Box–Behnken	12 edge midpoints	15

Totals include three runs at the center.

- A **central composite design** adds axial points such as $(\pm\alpha, 0, 0)$ to a two-level design and its center. At $\alpha = 1$ they sit at face centers.
- For the three-factor **Box–Behnken design**, run a 2^2 factorial for each pair of factors while holding the third at 0. Add center runs.

Box–Behnken for the forest

Factor	-1	0	$+1$
x_1 Features per split	1	5	9
x_2 Minimum leaf size	2	16	30
x_3 Maximum depth	2	5	8

For example, $(+1, -1, 0)$ uses 9 features, minimum leaf size 2, and depth 5.

Measure CV RMSE at the 12 edge midpoints. Run the center 3 times with different forest seeds and the same folds. Fit the quadratic by least squares.

§3. Steepest descent

The slope of a plane

For a first-order fit $\hat{m}(\boldsymbol{x}) = \hat{\beta}_0 + \boldsymbol{b}^\top \boldsymbol{x}$, a step of length h in a unit direction $\boldsymbol{d}$ changes the predicted response by

$$\hat{m}(\boldsymbol{x} + h\boldsymbol{d}) - \hat{m}(\boldsymbol{x}) = h\, \boldsymbol{b}^\top \boldsymbol{d}.$$

Which direction d makes this change as negative as possible?

The gradient of a quadratic

$$\hat{m}(\boldsymbol{x}) = \hat{\beta}_0 + \boldsymbol{b}^\top \boldsymbol{x} + \boldsymbol{x}^\top B \boldsymbol{x}, \qquad B = B^\top$$

The diagonal of B holds the square coefficients. Each off-diagonal entry is half the corresponding interaction coefficient.

$$\nabla \hat{m}(\boldsymbol{x}) = \boldsymbol{b} + 2B\boldsymbol{x}$$

The fitted direction changes as we move through the design region.

A step in coded units

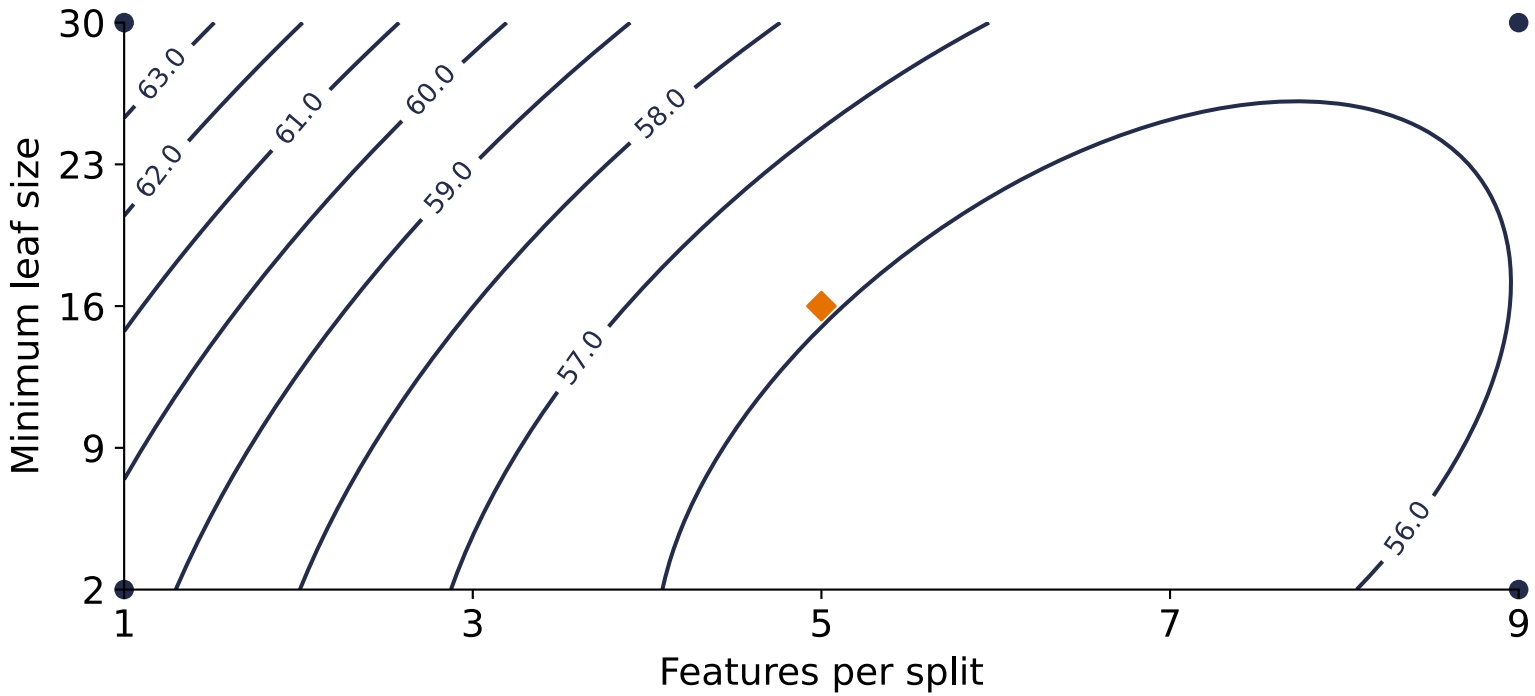
For a nonzero fitted gradient $g = \nabla \hat{m}(x)$ and a distance h,

$$\boxed{x_{\text{proposed}} = x - h\frac{g}{\| g \|}} \qquad \text{(descent)}\ .$$

- Use the plus sign for ascent.
- Keep the proposal in the allowed range and round to integer settings.
- Train and score the proposed forest. Continue from an improving setting.

A fitted slice for the forest

Depth is fixed at 5. Circles mark four design runs. The diamond marks the center.



The first step

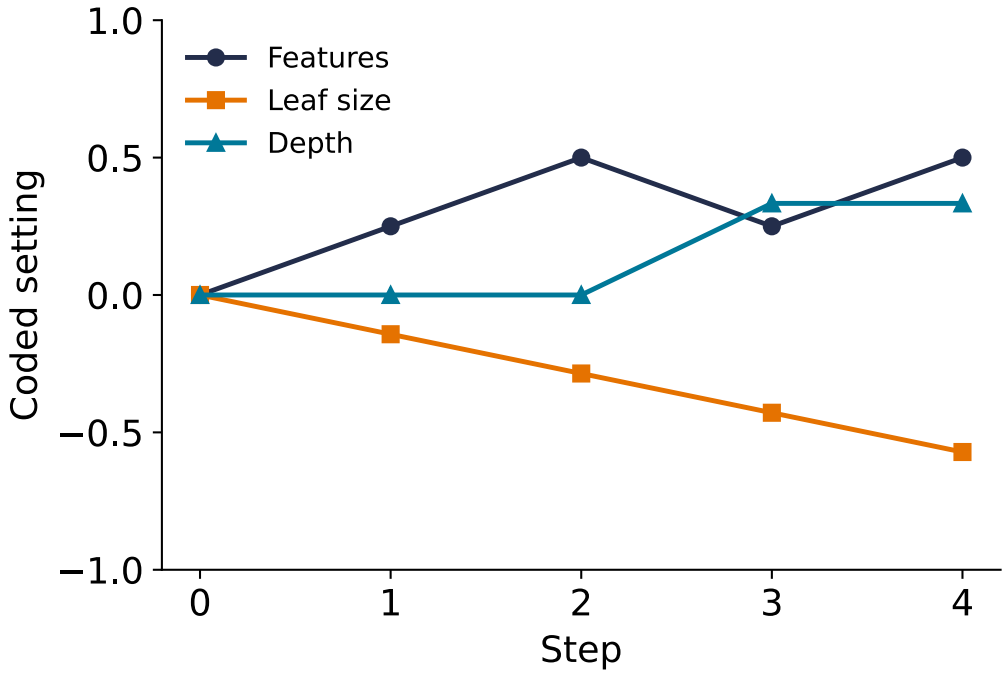
Start at $(0, 0, 0)$, the forest with settings $(5, 16, 5)$. Take $h = 0.35$.

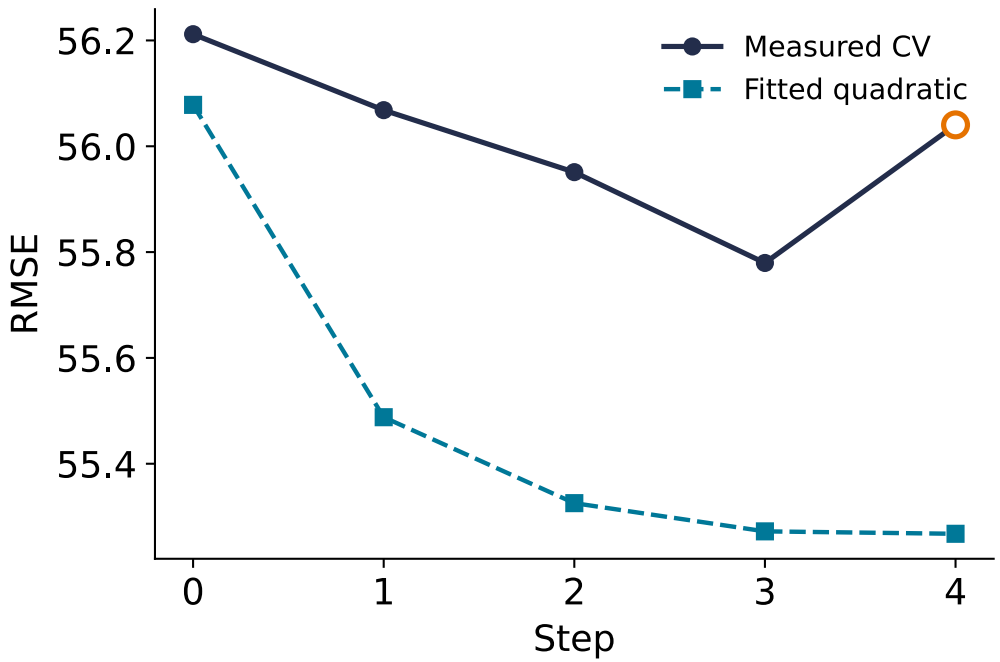
$$g = b = (-2.59, 1.11, -0.52), \qquad x_{\text{proposed}} = (0.32, -0.14, 0.06)$$

After decoding and rounding, use $(6, 14, 5)$ in the forest.

For the next step, evaluate the gradient at the **coded settings actually run**.

Measured error along the path





Stop when measured RMSE stops falling. A further experiment could shrink and recenter the region around a promising setting.

Checking the selected settings

Choose the lowest CV RMSE among the settings tested under the common seed. Refit on all 331 tuning observations and evaluate on the 111 test observations.

Forest	(Features, leaf size, depth)	CV RMSE	Test RMSE
Starting point	$(5, 16, 5)$	56.21	58.25
Selected setting	$(6, 10, 6)$	55.78	57.35

Where would you put the next group of runs?

Reading

- Lujan-Moreno et al. (2018), random-forest case study.
- Wu and Hamada, *Experiments*, 3rd ed., §§10.2–10.4 and §§10.7–10.8.
- Owen, *A First Course in Experimental Design*, §§12.1–12.5.

Appendix

Scoring a setting

```
import numpy as np
from sklearn.datasets import load_diabetes
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import KFold, cross_val_predict, train_test_split

X, y = load_diabetes(return_X_y=True, scaled=False)
X_tune, X_test, y_tune, y_test = train_test_split(
    X, y, test_size=0.25, random_state=6160
)
folds = list(KFold(5, shuffle=True, random_state=6161).split(X_tune))
forest = RandomForestRegressor(
    n_estimators=96, max_features=5, min_samples_leaf=16, max_depth=5,
    random_state=6162, n_jobs=1,
)
predictions = cross_val_predict(forest, X_tune, y_tune, cv=folds)
rmse = np.sqrt(np.mean((y_tune - predictions)**2))
```

Fitting and stepping

```
design = box_behnken()
seeds = [6162]*13 + [6163, 6164]
scores = [cv_rmse(decode(x), seed) for x, seed in zip(design, seeds)]
beta = np.linalg.lstsq(quadratic_matrix(design), scores, rcond=None)[0]
b = beta[1:4]
b12, b13, b23 = beta[4:7]
B = np.diag(beta[7:])
B += np.array([[0, b12, b13], [b12, 0, b23], [b13, b23, 0]]) / 2

x = np.zeros(3)
gradient = b + 2 * B @ x
proposal = decode(x - 0.35 * gradient / np.linalg.norm(gradient))
cv_rmse(proposal, 6162)
```

The Quarto source contains the design generator and the full descent loop.