Hyperparameters and Validation

K Arnold

```r
data(ames, package = "modeldata")
ames_all <- ames %>%
  filter(Gr_Liv_Area < 4000, Sale_Condition == "Normal") %>%
  mutate(across(where(is.integer), as.double)) %>%
  mutate(Sale_Price = Sale_Price / 1000)
rm(ames)
```

```r
set.seed(10) # Seed the random number generator
ames_split <- initial_split(ames_all, prop = 2 / 3)
ames_train <- training(ames_split)
ames_test <- testing(ames_split)
```

```r
model1 <-
  decision_tree(mode = "regression", tree_depth = 2) %>%
  fit(Sale_Price ~ Latitude + Longitude, data = ames_train)
```

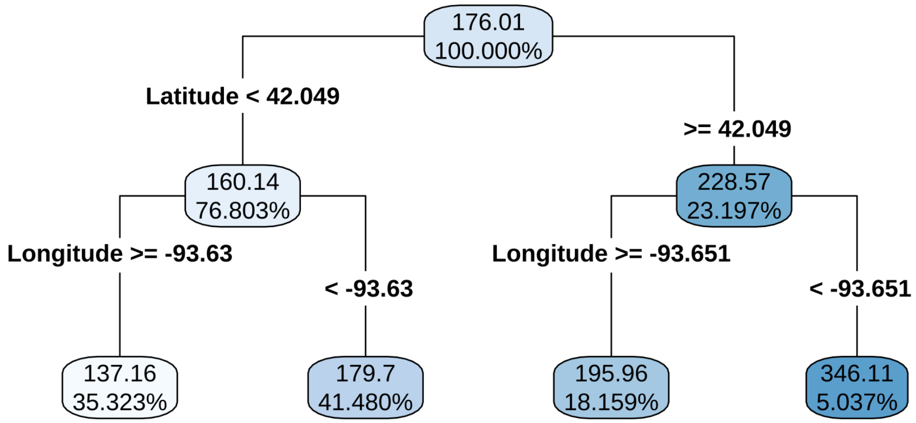
```r
model2 <-
  decision_tree(mode = "regression", tree_depth = 30) %>%
  fit(Sale_Price ~ Latitude + Longitude, data = ames_train)
```

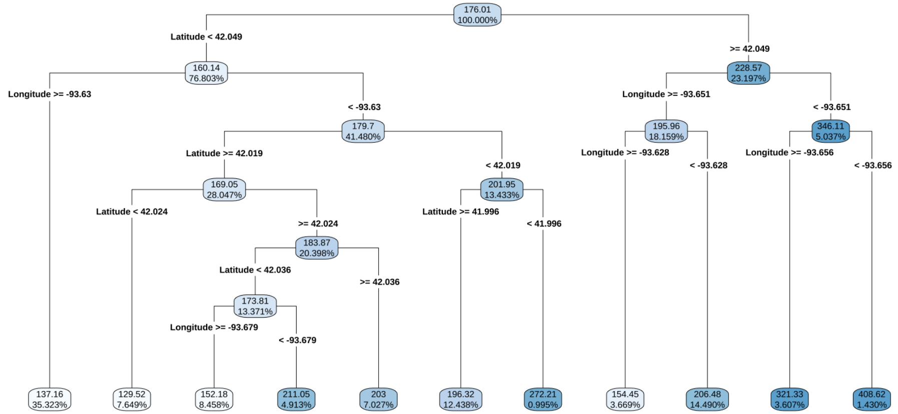
```r
model3 <-
  decision_tree(mode = "regression", cost_complexity = 1e-6, min_n = 2) %>%
  fit(Sale_Price ~ Latitude + Longitude, data = ames_train)
```

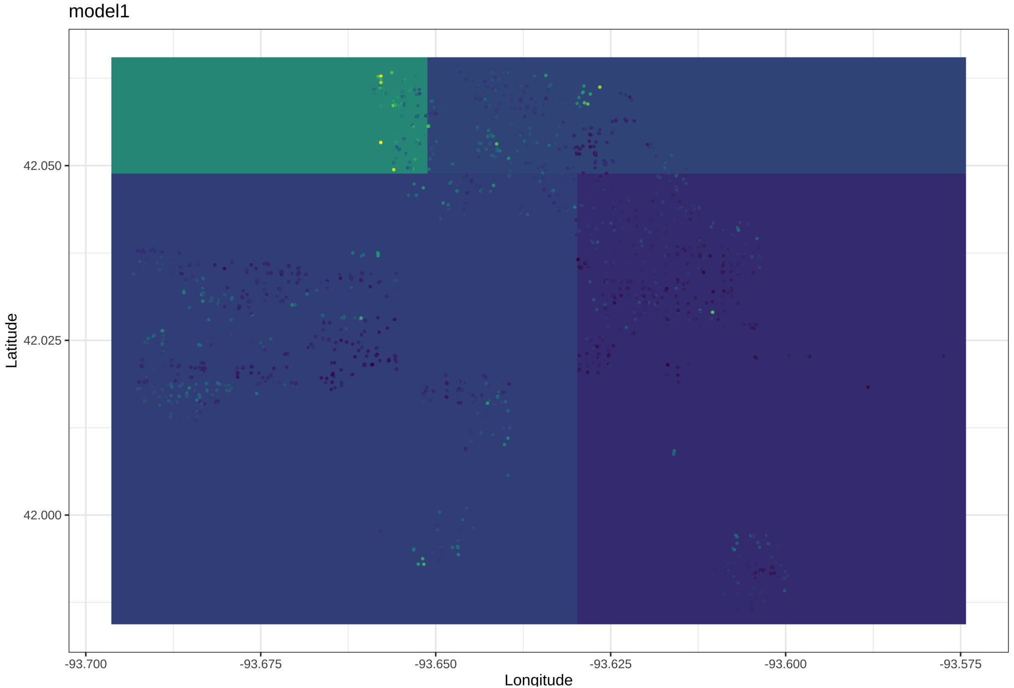
This week

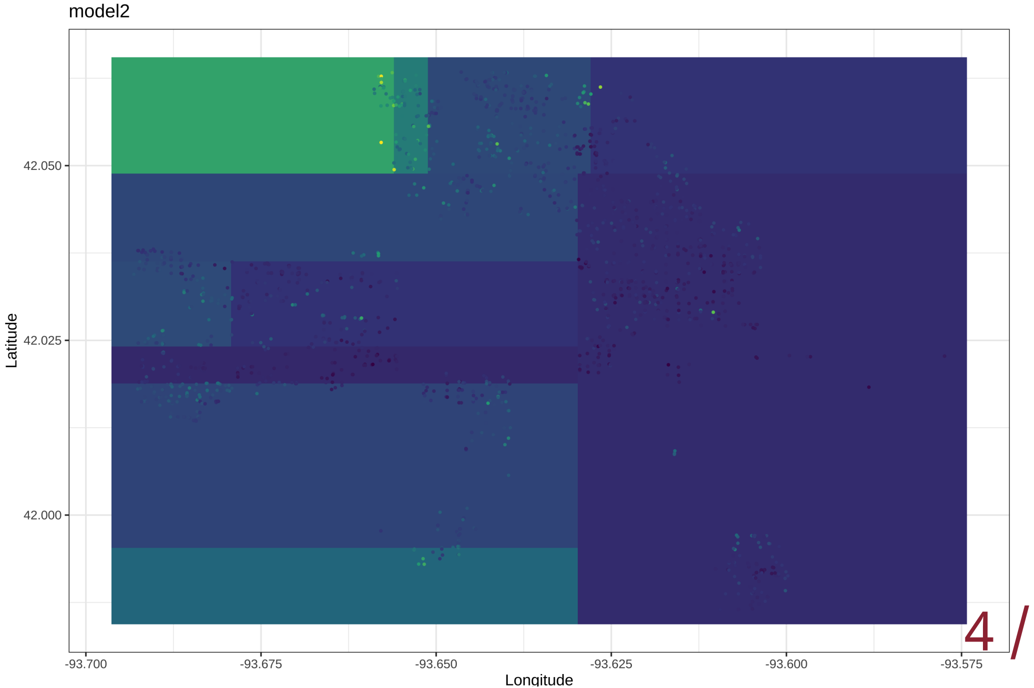
- *Today*: how modeling decisions affect performance; why validate?
- *Wednesday*: **how** to validate?
- *Friday*: cross-validation lab

Decision Trees









Q&A

> How do we *train* a decision tree?

- The model: "choose your own adventure": at each step, check one simple condition about one variable (e.g., `Latitude < 42.05`)
- Goal: find the best tree (for regression: minimize MSE)
- Approach: greedy algorithm: try all possible splits, keep the best one, repeat.

> I missed one of the check-in quizzes (or even weekly quizzes)!

Email me.

> Is midterm project individual?

At this point, yes.

Objectives

- Identify modeling decisions that affect the performance of decision tree and linear regression models, including:
    - Choice of model type
    - Pre-processing steps
    - Hyperparameter settings
- Explain the importance of *validation* for assessing and comparing models.

What are some decisions that we've made when making models so far?

What are some decisions that we've made when making models so far?

- Which model to use

What are some decisions that we've made when making models so far?

- Which model to use
- Shifting and scaling features

What are some decisions that we've made when making models so far?

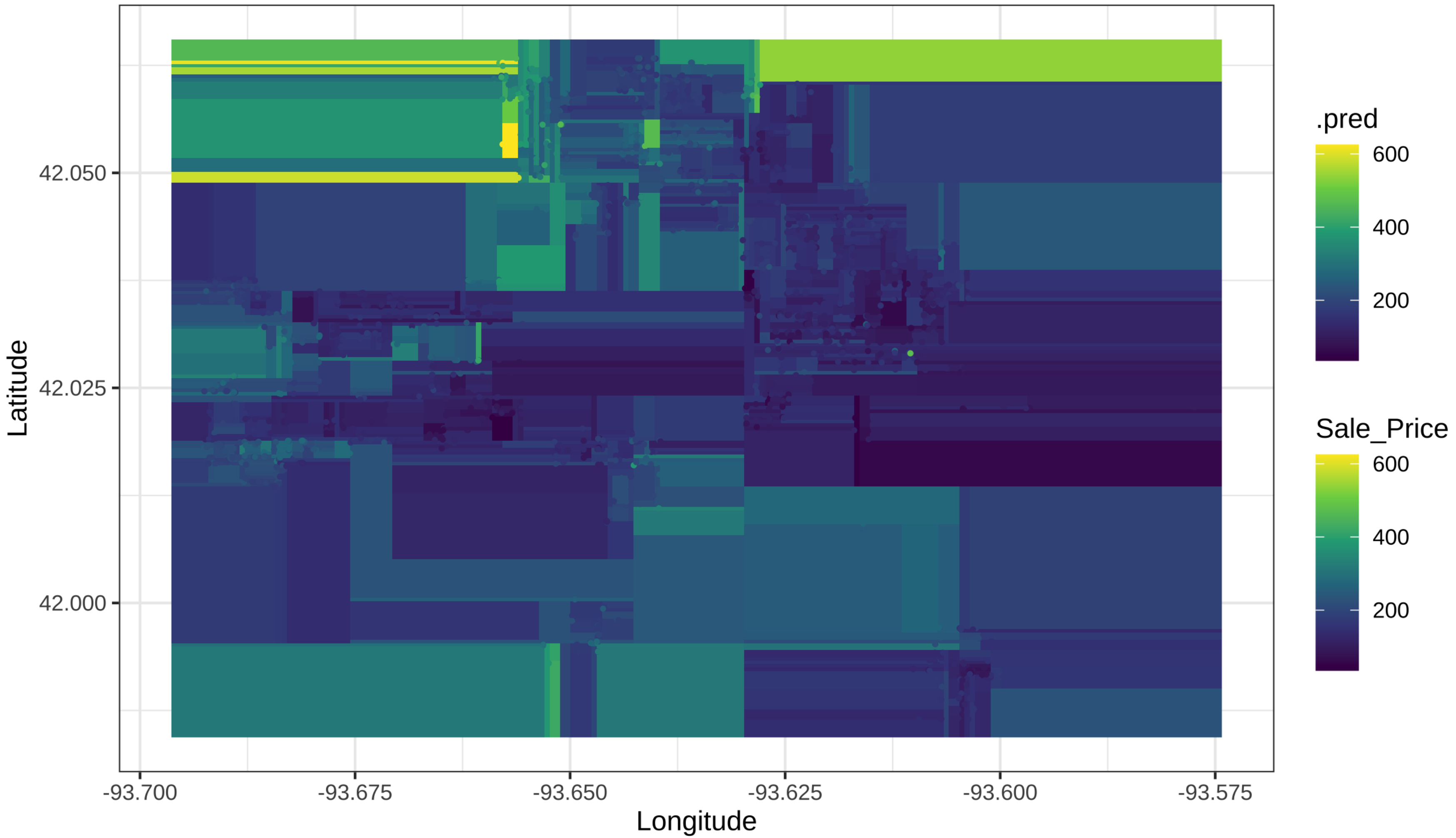
- Which model to use
- Shifting and scaling features
- Hyperparameters: Tree depth, number of observations per leaf, ...

Hyperparameters for Decision Trees

```
model1 <-
  decision_tree(mode = "regression", tree_depth = 2) %>%
  fit(Sale_Price ~ Latitude + Longitude, data = ames_train)
model2 <-
  decision_tree(mode = "regression", tree_depth = 30) %>%
  fit(Sale_Price ~ Latitude + Longitude, data = ames_train)
model3 <-
  decision_tree(mode = "regression", cost_complexity = 1e-6, min_n = 2) %>%
  fit(Sale_Price ~ Latitude + Longitude, data = ames_train)
```

- Tree depth: how many levels of decisions
- Leaf size: how many observations need to be in each leaf node
- Complexity penalty: how much improvement for a split to be "worth it"

model3

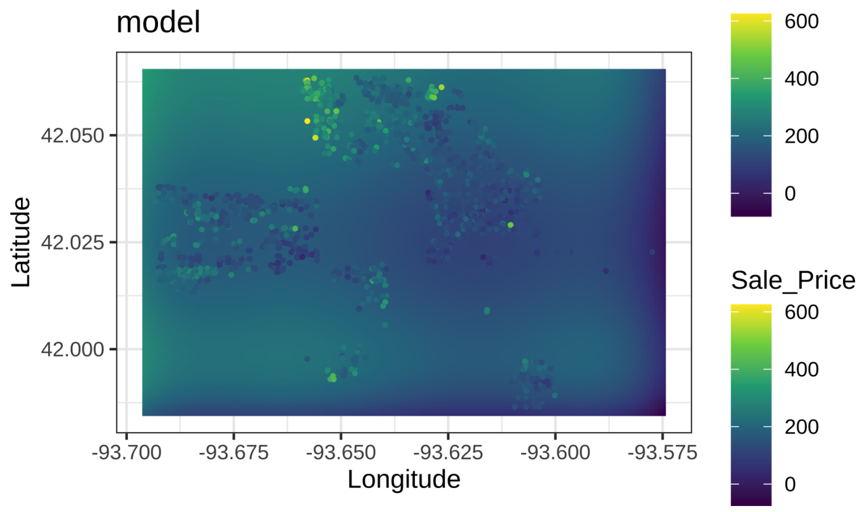


42.050
42.025
42.000
Latitude
-93.700
-93.675
-93.650
-93.625
-93.600
-93.575
Longitude
.pred
600
400
200
Sale_Price
600
400
200

Hyperparameters for Linear Regression

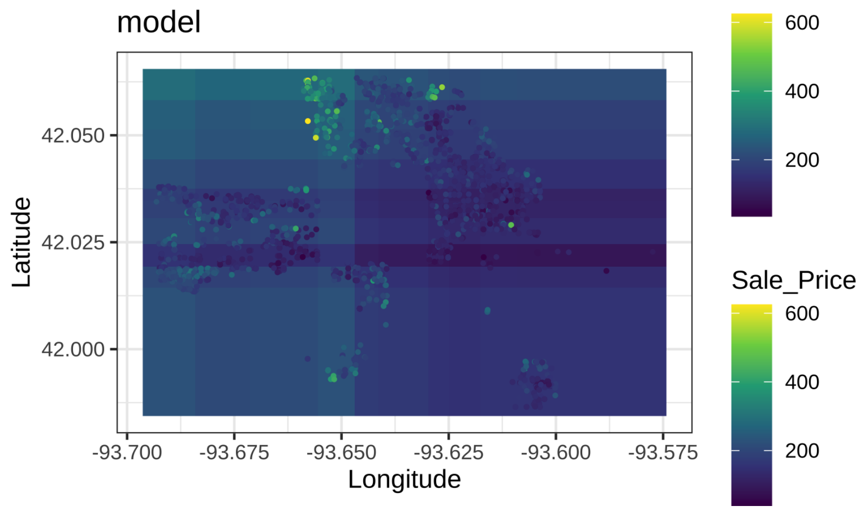
```r
recipe <-
  recipe(Sale_Price ~ Latitude + Longitude, da
  step_poly(Latitude, Longitude, degree = 5)
```

```r
model <- workflow() %>% add_recipe(recipe) %>% add_model(linear_reg()) %>%
  fit(ames_train)
show_latlong_model(ames_train, model)
```




```r
recipe <-
  recipe(Sale_Price ~ Latitude + Longitude, da
  step_discretize(Latitude, Longitude, num_bre
```

```r
model <- workflow() %>% add_recipe(recipe) %>% add_model(linear_reg()) %>%
  fit(ames_train)
show_latlong_model(ames_train, model)
```



Polynomial and interaction features

```r
recipe <-
  recipe(Sale_Price ~ Gr_Liv_Area + Bldg_Type,
  step_dummy(Bldg_Type) %>%
  step_interact(
    ~ Gr_Liv_Area:starts_with("Bldg_Type")) %>%
  step_poly(
    starts_with("Gr_Liv_Area"), degree = 2)
```

```r
recipe <-
  recipe(Sale_Price ~ Gr_Liv_Area + Bldg_Type,
  step_dummy(Bldg_Type) %>%
  step_interact(
    ~ Gr_Liv_Area:starts_with("Bldg_Type")) %>%
  step_poly(
    starts_with("Gr_Liv_Area"), degree = 5)
```