Hyperparameters and Cross Validation

DATA 202 21FA

```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
metrics <- yardstick::metric_set(mae, mape, rsq_trad)

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)
```

Q&A

> Where to look for final project ideas? How modeling is used?

Pick a topic, search for "*topic* Kaggle", "*topic* data science blog" etc.; more ideas on the Projects page.

> What other areas does sensitivity/specificity matter for?

Medical tests. Fraud detection. Content moderation. Lots!

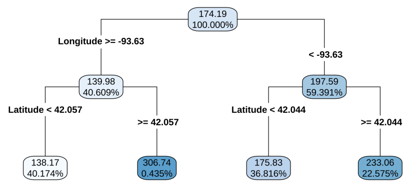
> Can we do final project in teams?

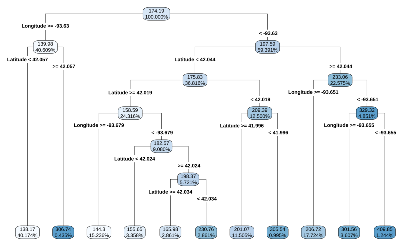
Yes, even across sections. See milestone instructions.

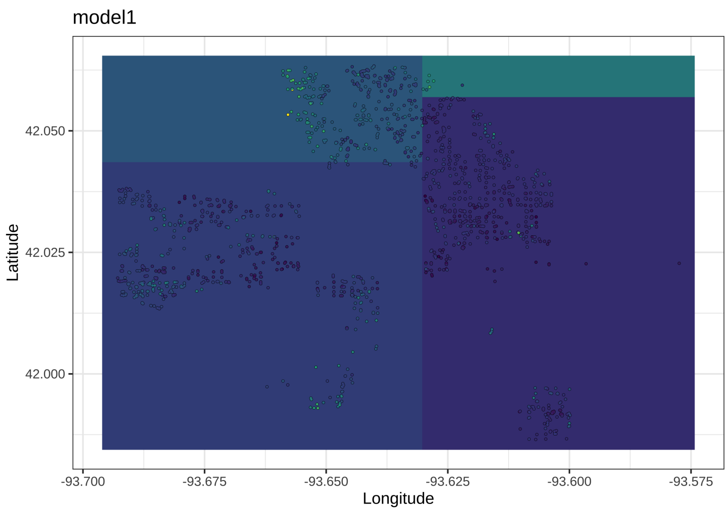
Location, Location, Location!

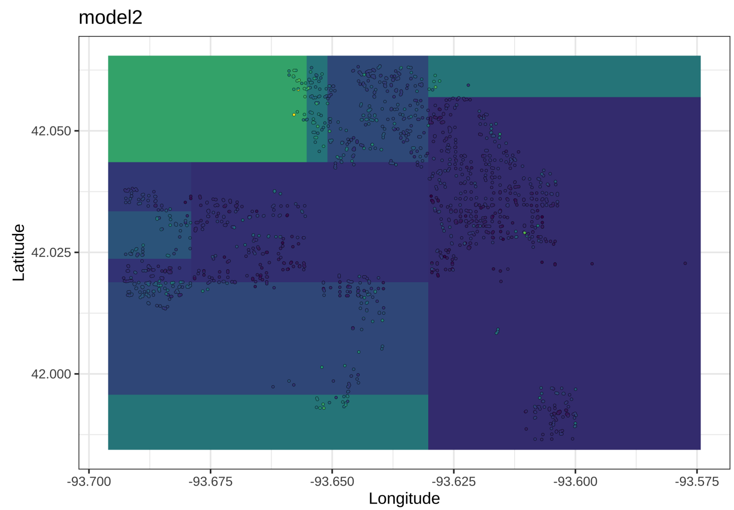
- Recall: Ames housing dataset has sale prices for homes. **Task**: Predict how much a home will sell for.
- Previously we used attributes of the house and lot.
- Today (just to illustrate), we'll look at location *only*.

Which model is better?

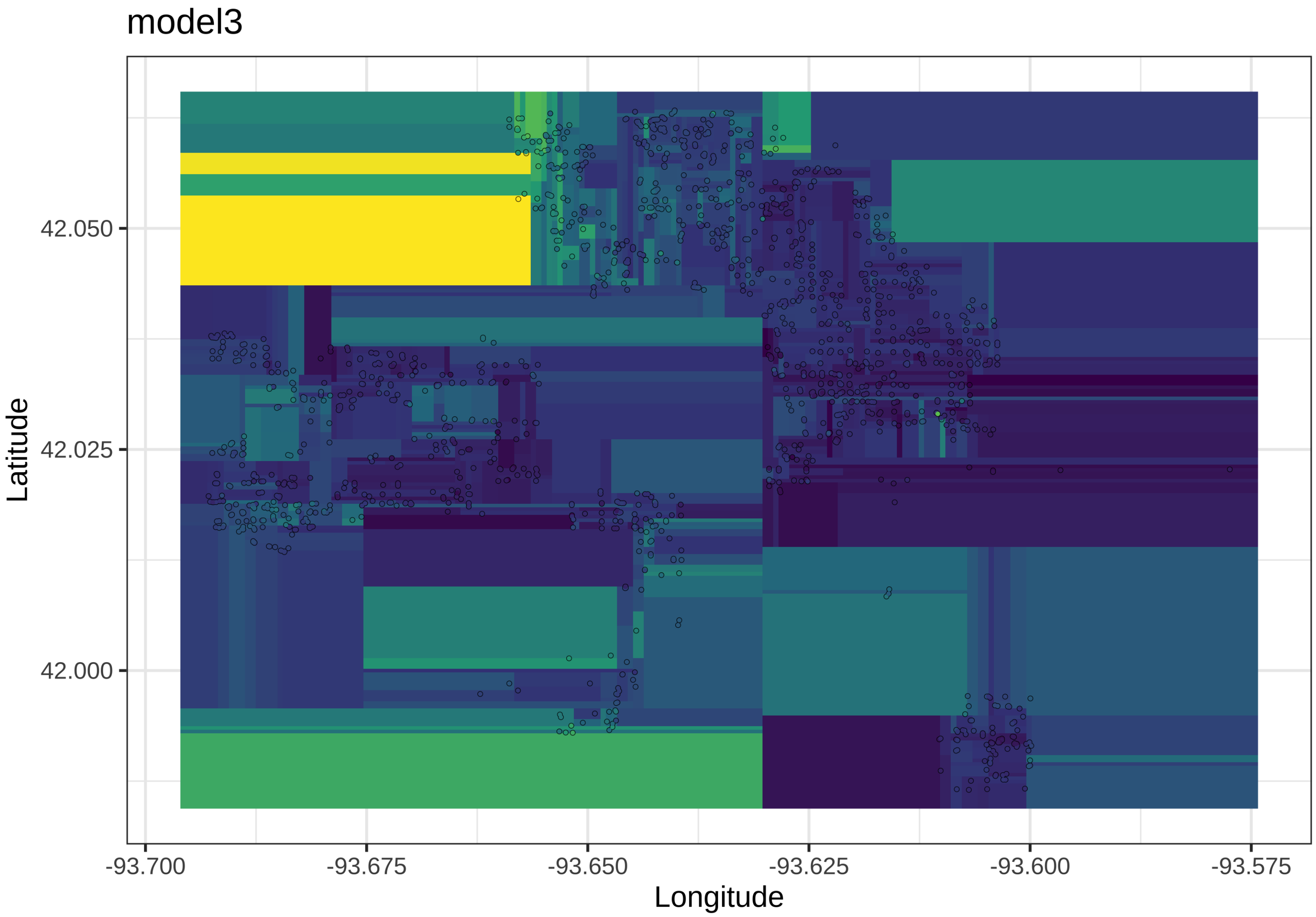








How about this model?



What made these models different? *Hyperparameters*

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

```
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
  fit(Sale_Price ~ Latitude + Longitude, data = ames_train)
```

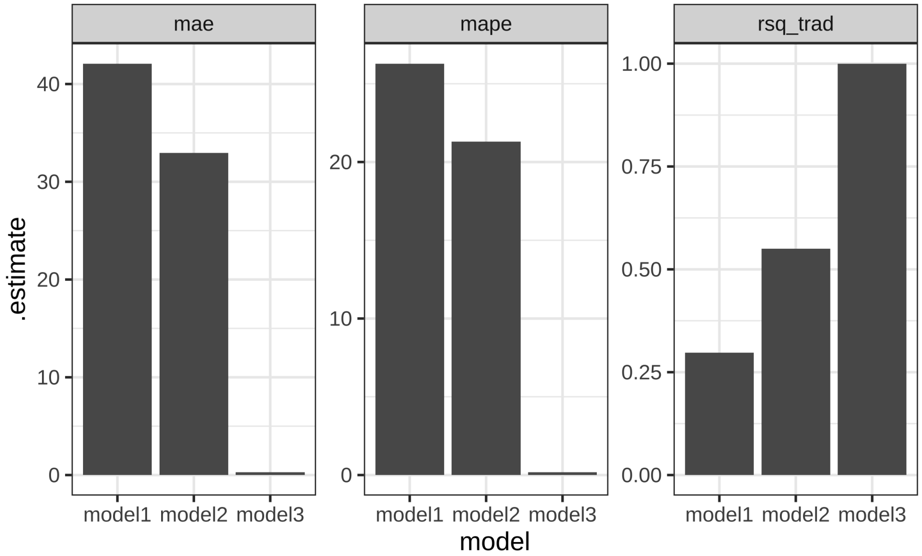
How do we *train* a decision tree?

Greedy algorithm: make the best single split of the current data, repeat.

- 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.

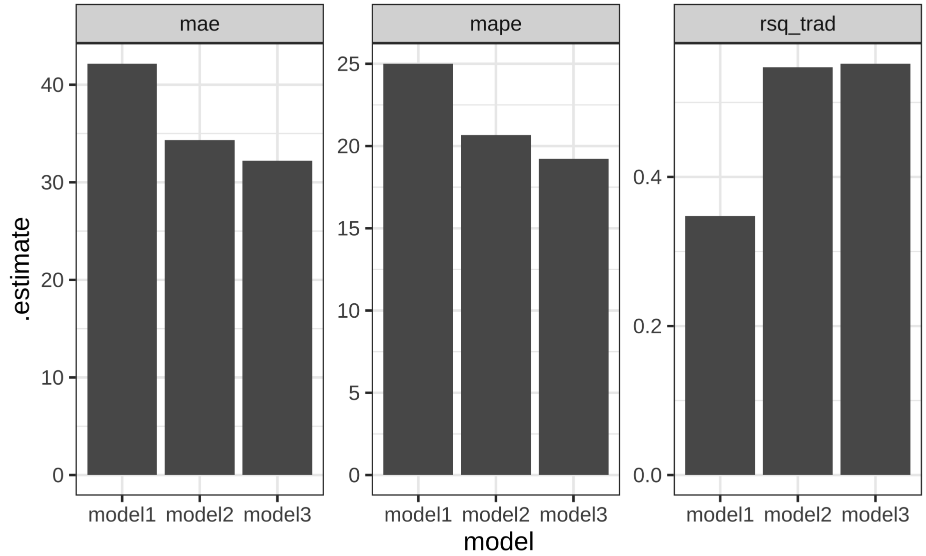
Which one works best?

```
bind_rows(
  ames_train %>% add_predictions(model1),
  ames_train %>% add_predictions(model2),
  ames_train %>% add_predictions(model3),
) %>%
  group_by(model) %>%
  metrics(truth = Sale_Price, estimate = .pred) %>%
  ggplot(aes(y = .estimate, x = model)) + geom_col() +
  facet_wrap(vars(.metric), scales = "free_y")
```

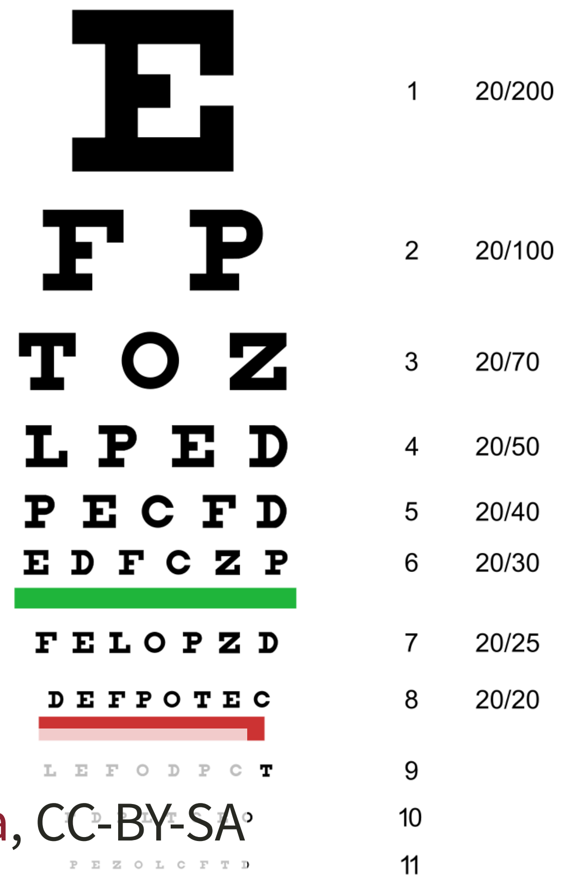


How about on testing data?

```
bind_rows(
  ames_test %>% add_predictions(model1),
  ames_test %>% add_predictions(model2),
  ames_test %>% add_predictions(model3),
) %>%
  group_by(model) %>%
  metrics(truth = Sale_Price, estimate = .pred) %>%
  ggplot(aes(y = .estimate, x = model)) + geom_col() +
  facet_wrap(vars(.metric), scales = "free_y")
```



Why train-test split? Memorizing the eye chart



Snellen chart on Wikimedia,
Analogy by Clem Wang

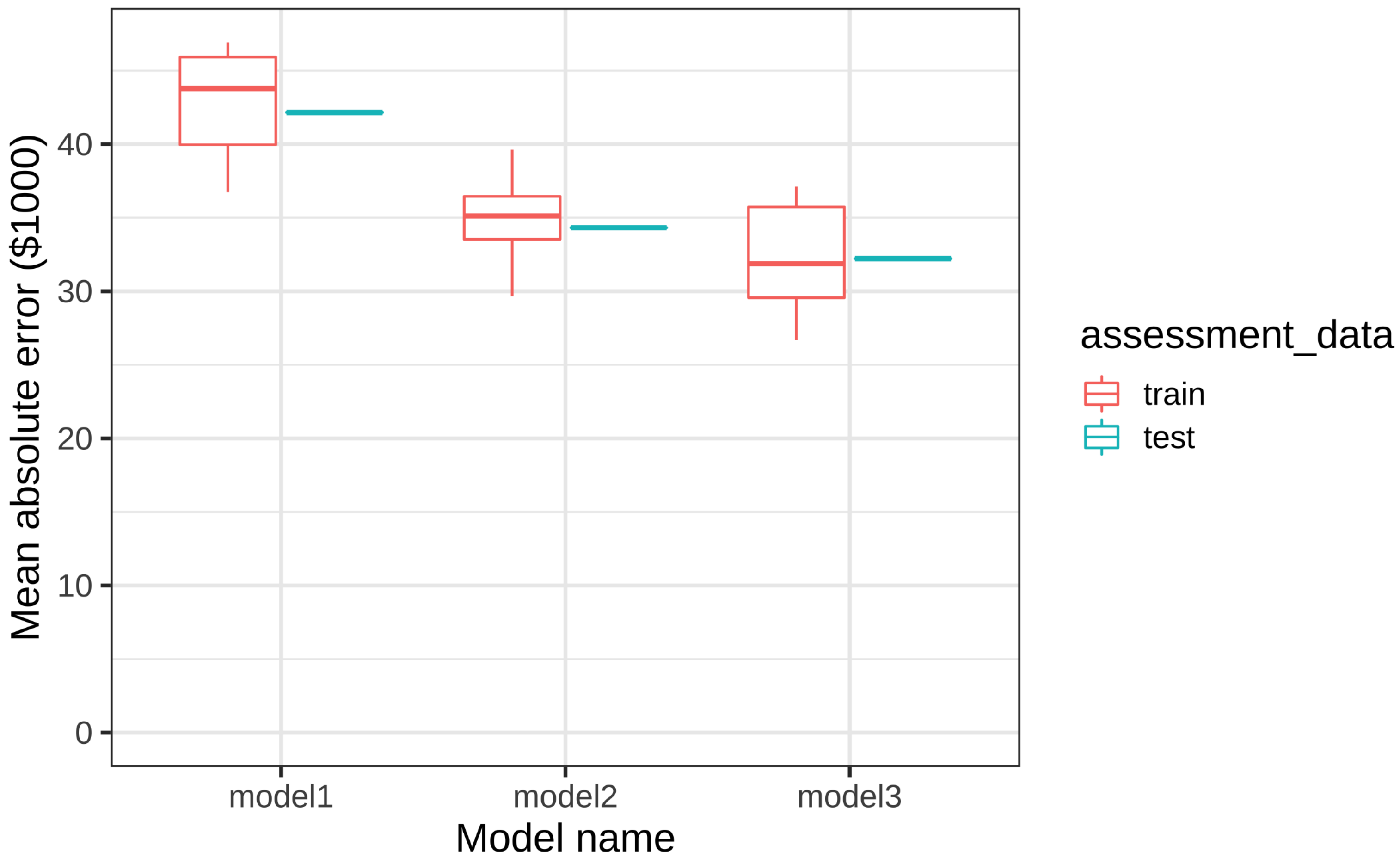
Cross-Validation

Puzzle

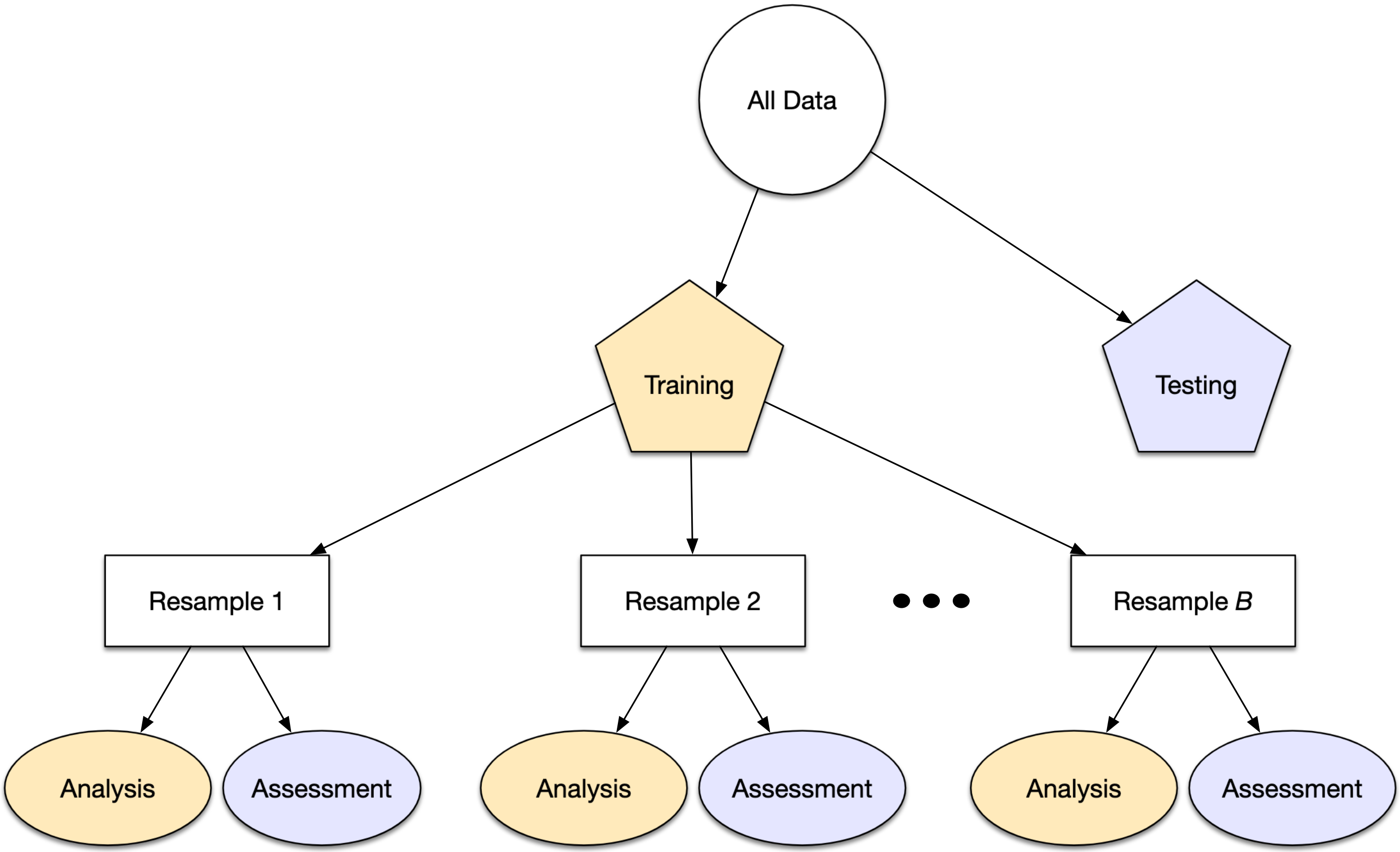
- We want to pick the model that works best on *unseen* data
- ... but as soon as we try one model, **we've peeked at the data!**

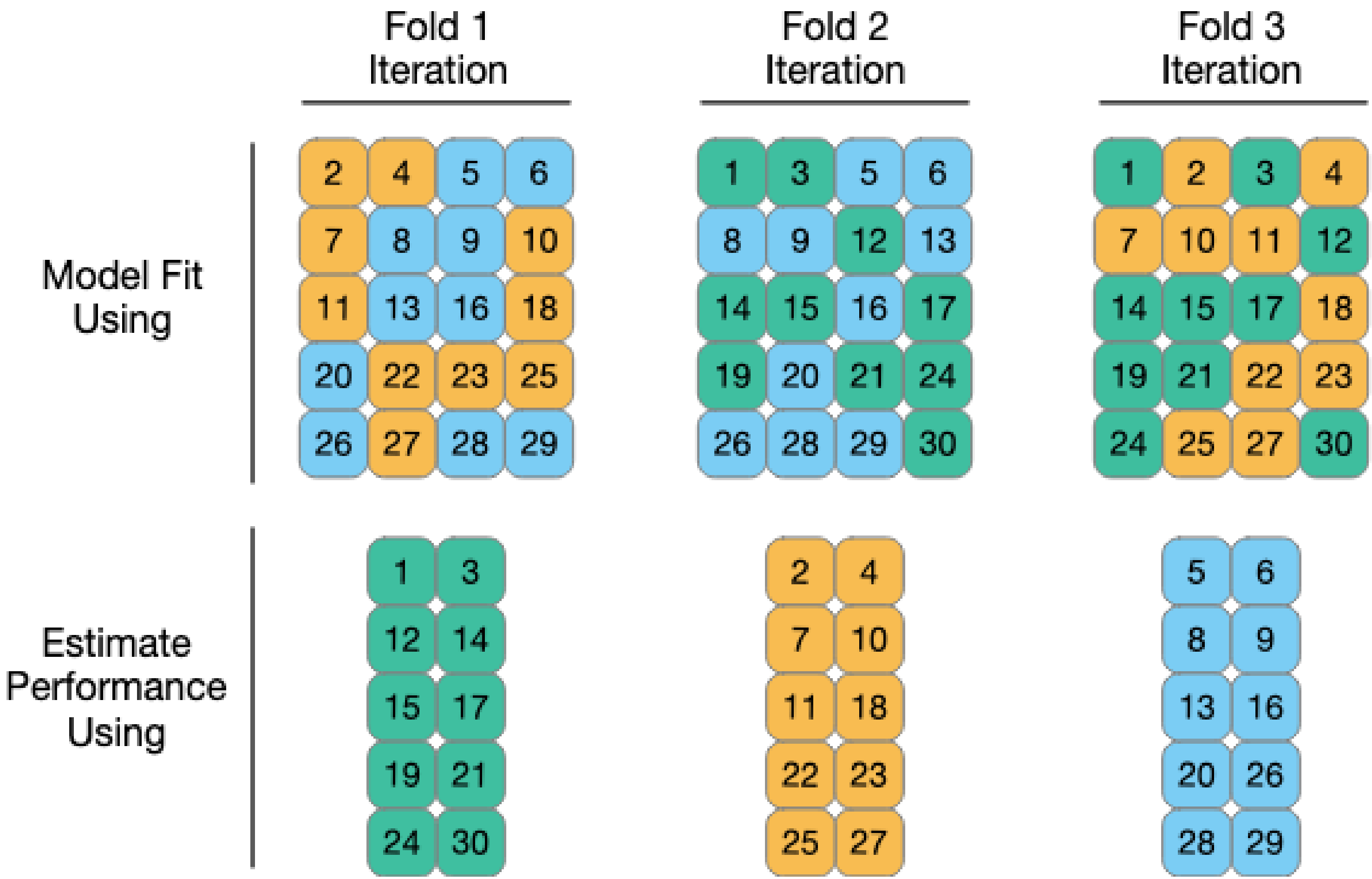
Solution

- Divide training data into V piles (e.g., 10)
- Hide one pile from yourself.
  - train on ("analyze") the rest,
  - evaluate ("assess") on the one you held out.
- Repeat for each of the V piles.

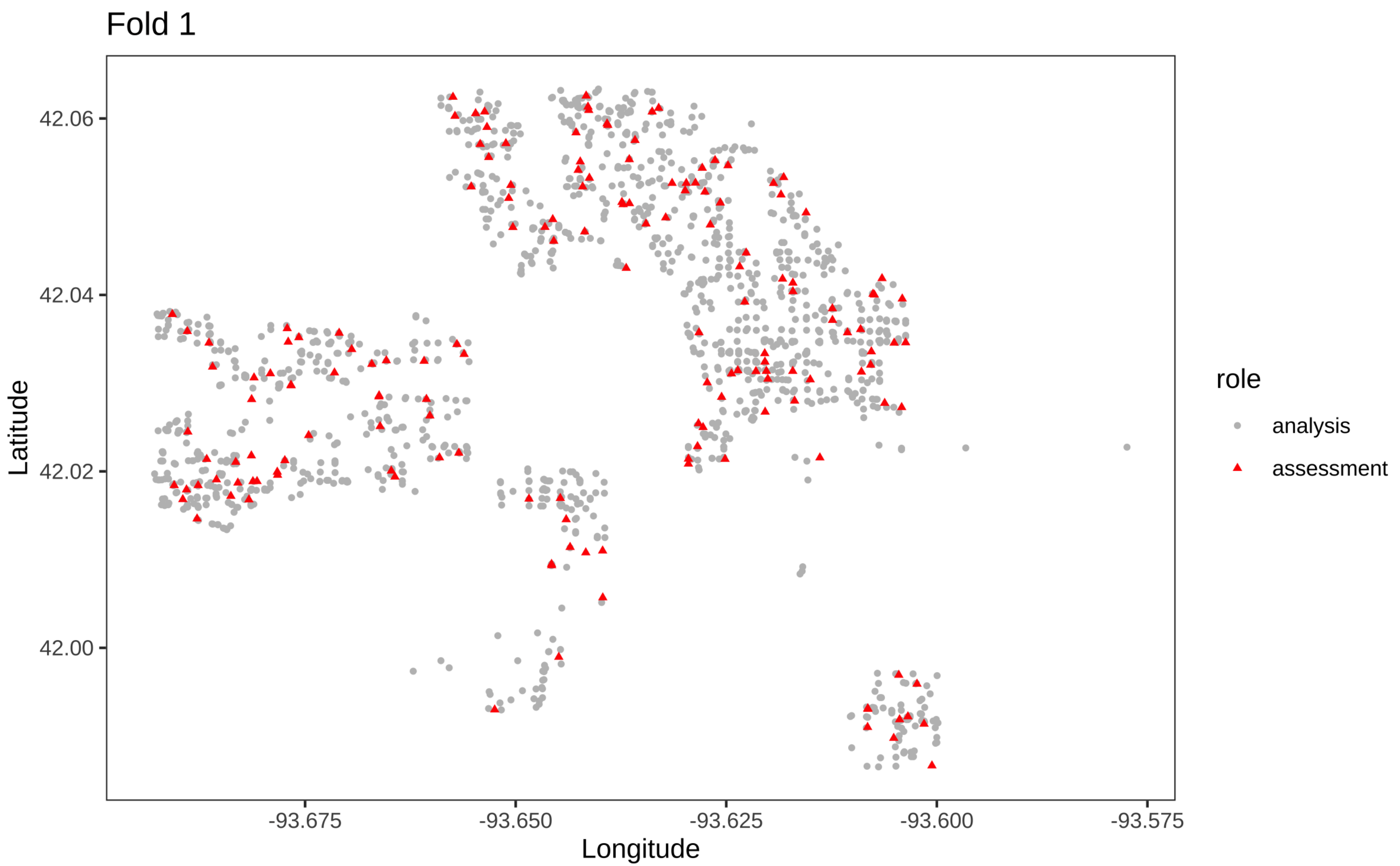

Mean absolute error ($1000)
40
30
20
10
0
model1
model2
model3
Model name
assessment_data
train
test

What is Cross-Validation?





Source: Tidy Modeling with R



Fold 1
Latitude
42.06
42.04
42.02
42.00
-93.675
-93.650
-93.625
-93.600
-93.575
Longitude
role
analysis
assessment

How to do CV?

1. Declare the splitting strategy:

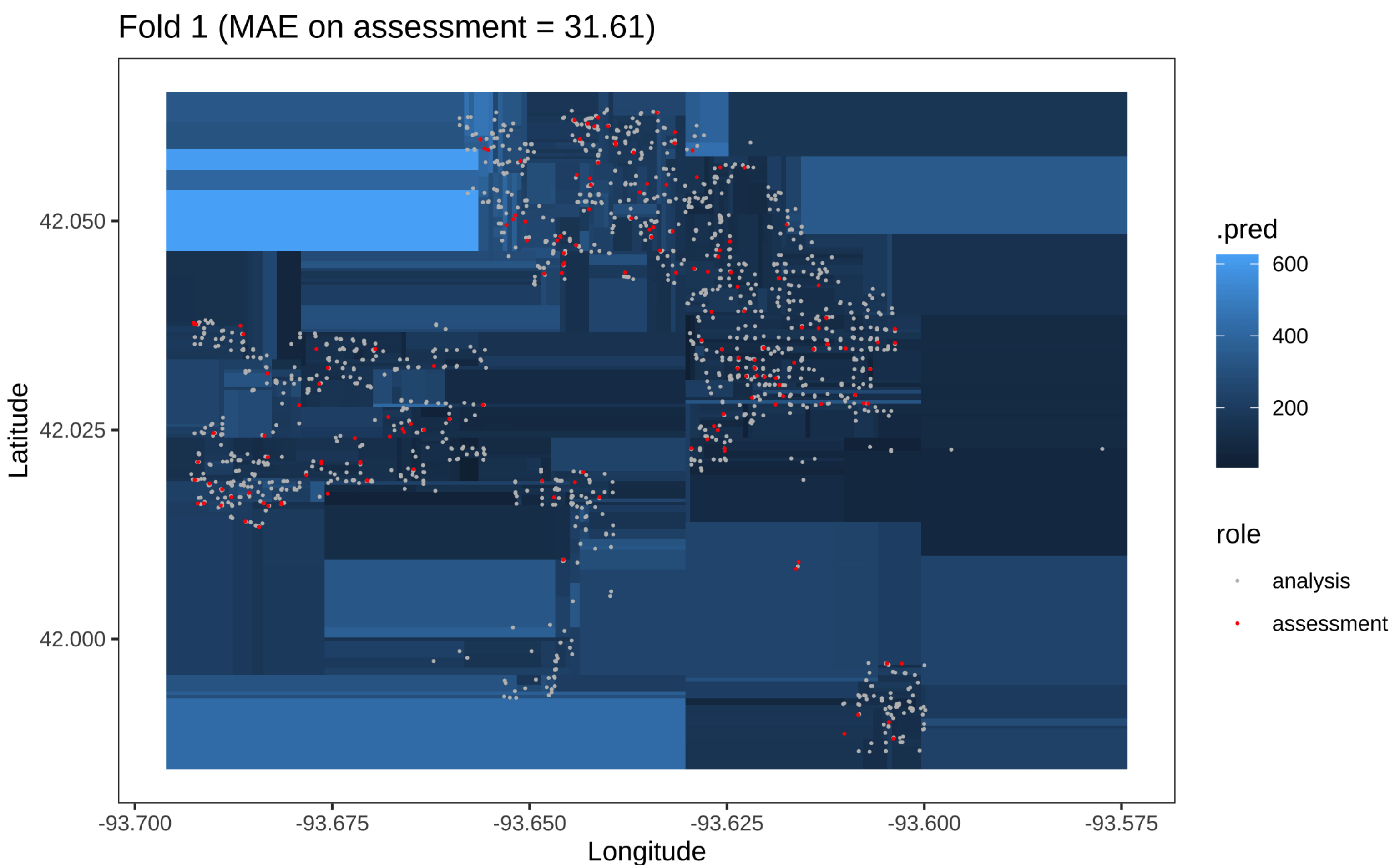
```
ames_resamples <- ames_train %>% vfold_cv(v = 10)
```

```
ames_resamples
```

```
#  10-fold cross-validation
# A tibble: 10 × 2
  splits             id
  <list>             <chr>
1 <split [1447/161]> Fold01
2 <split [1447/161]> Fold02
3 <split [1447/161]> Fold03
4 <split [1447/161]> Fold04
5 <split [1447/161]> Fold05
6 <split [1447/161]> Fold06
# … with 4 more rows
```

How to do CV?

1. Declare the splitting strategy
2. Fit on each resample, evaluate using a set of metrics.



How to do CV?

1. Declare the splitting strategy
2. Fit on each resample, evaluate using a set of metrics.

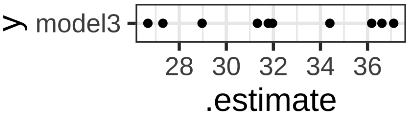
```
model3_samples <- model3_spec %>%
  fit_resamples(
    Sale_Price ~ Latitude + Longitude,
    resamples = ames_resamples,
    metrics = metric_set(mae))
model3_samples %>% collect_metrics(summarize = FALSE)
```

```
# A tibble: 10 × 5
  id     .metric .estimator .estimate .config
  <chr>  <chr>   <chr>          <dbl> <chr>
1 Fold01 mae     standard        29.0 Preprocessor1_Model1
2 Fold02 mae     standard        36.6 Preprocessor1_Model1
3 Fold03 mae     standard        34.4 Preprocessor1_Model1
4 Fold04 mae     standard        31.8 Preprocessor1_Model1
5 Fold05 mae     standard        32.0 Preprocessor1_Model1
6 Fold06 mae     standard        27.3 Preprocessor1_Model1
# … with 4 more rows
```

How to do CV?

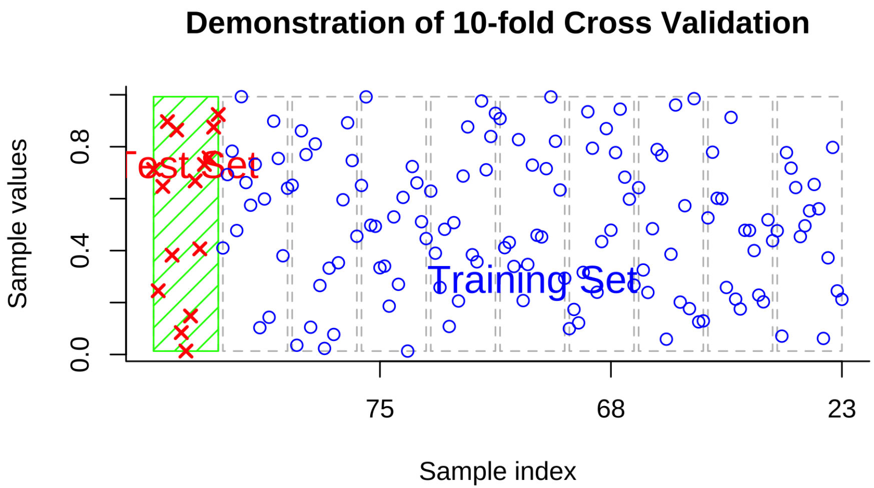
1. Declare the splitting strategy
2. Fit on each resample, evaluate using a set of metrics.
3. Plot and/or summarize the metrics.

```
model3_samples %>%
  collect_metrics(summarize =
  ggplot(aes(x = .estimate, y
  geom_point()
```




```
model3_samples %>%
  collect_metrics(summarize =
```

```
# A tibble: 1 × 6
  .metric .estimator  mean
  <chr>   <chr>      <dbl> <int
1 mae     standard    32.2     1
```


Demonstration of 10-fold Cross Validation
Sample values
0.0
0.4
0.8
Test Set
Training Set
75
68
23
Sample index

In code...

```
cross_val_scores <- function(complete_model_spec, training_data,
  # Split the data into V folds.
  set.seed(0)
  resamples <- vfold_cv(training_data, v = v)

  ...
}
```

In code...

```r
cross_val_scores <- function(complete_model_spec, training_data,
  # Split the data into V folds.
  set.seed(0)
  resamples <- vfold_cv(training_data, v = v)

  # For each of the V folds, assess the result of analyzing on th
  raw_cv_results <- complete_model_spec %>%
    fit_resamples(resamples = resamples, metrics = metrics)

  # Return the collected metrics.
  collect_metrics(raw_cv_results, summarize = FALSE)
}
```

What's a complete model spec?

Workflow = recipe + model_spec.

```
spec <- workflow() %>%
  add_recipe(recipe) %>%
  add_model(model)
```

e.g.,

```
spec <- workflow() %>%
  add_recipe(
    recipe(Sale_Price ~ Latitude + Longitude, data = ames_train)
  ) %>%
  add_model(
    linear_reg()
  )
```