What makes a ‘good’ model?

DATA 202 21FA

Q&A

> What other kinds of models can we use?

- We'll also study *linear regression* (and its extensions) and *logistic regression* (which is a *classification* model despite the name).
- Hot right now: Neural Networks

> Can we predict whether homes will get more expensive or not?

That would be *forecasting*, one type of prediction. Tricky to do right because you're *extrapolating*.

Objectives

- Compare and contrast regression tasks and classification tasks, and give examples of each
- Identify two different ways of measuring accuracy for regression and for classification
- Identify several reasons why a model may predict better on some subsets of data than others

Types of Tasks

- **regression**: predict a *number* ("continuous")
  - number should be "close" in some sense to the correct number
- **classification**: predict a *category*
  - which one of these two groups? three groups? 500,000 groups?
  - could ask: "how likely is it to be in group *i*"

Are these tasks *regression* or *classification*?

1. Is this a picture of the inside or outside of the restaurant?
2. How much will it rain in GR next year?
3. Is this person having a seizure?
4. How much will this home sell for?
5. How much time will this person spend watching this video?
6. How big a fruit will this plant produce?
7. Which word did this person mean to type?
8. Will this person "Like" this post?

Today's examples

Regression: housing prices in Ames, Iowa. Details:

- Paper
- Data Dictionary

What makes a good prediction? *Regression*

We predicted the home would sell for $250k. It sold for $200k. Is that good?

What makes a good prediction? *Regression*

We predicted the home would sell for $250k. It sold for $200k. Is that good?

- **residual**: actual minus predicted
  If home sold for $200k but we predicted $250k, residual is __
- **absolute error**
- **squared error**

What makes a good prediction?
Regression

We predicted the home would sell for $250k. It sold for $200k. Is that good?

- **residual**: actual minus predicted
  If home sold for $200k but we predicted $250k, residual is __
- **absolute error**
- **squared error**

Across the entire dataset:

- **average error**: do we tend to predict too high? too low? "*bias*"
- **mean** or **max** absolute error
- **mean squared error** (MSE)
- normalized squared error: MSE / Variance
  - "R2" = 1 - normalized squared error

Most commonly used regression metrics

- MAE: Mean Absolute Error ("predictions are usually off by $xxx")
- MAPE: Mean Absolute Percent Error ("predictions are usually off by yy%")

And some math-y ones:

- Traditional R^2 (fraction of variance explained)
- RMSE: Root Mean Squared Error: sorta like the standard deviation

Seizure classification

First FDA-approved AI-powered medical device: Empatica Embrace2, company founded by MIT data scientist Rosalind Picard

What makes a good prediction? *Classification*

Suppose: every minute, the armband decides whether a seizure is occurring

The child was perfectly fine but our armband flagged a seizure. Is that good?

What makes a good prediction? *Classification*

Suppose: every minute, the armband decides whether a seizure is occurring

The child was perfectly fine but our armband flagged a seizure. Is that good?

The child was having a seizure but our armband didn't flag it. Is that good?

What makes a good prediction?
Classification

	Seizure predicted	**No seizure predicted**
Seizure happened	True positive	False negative (Type 1 error)
No seizure happened	False positive (Type 2 error)	True negative

What makes a good prediction? *Classification*

	Seizure predicted	**No seizure predicted**
Seizure happened	True positive	False negative (Type 1 error)
No seizure happened	False positive (Type 2 error)	True negative

- **Accuracy** (% correct) = (TP + TN) / (# predictions made)
- **False negative** ("miss") **rate** = FN / (# actual seizures)
- **False positive** ("false alarm") **rate** = FP / (# true non-seizures)

What makes a good prediction?
Classification

	Seizure predicted	No seizure predicted
Seizure happened	True positive	False negative (Type 1 error)
No seizure happened	False positive (Type 2 error)	True negative

- **Accuracy** (% correct) = (TP + TN) / (# predictions made)
- **False negative** ("miss") **rate** = FN / (# actual seizures)
- **False positive** ("false alarm") **rate** = FP / (# true non-seizures)
- **Sensitivity** ("true positive rate") = TP / (# true seizures)
  - Sensitivity = 1 − False negative rate
- **Specificity** ("true negative rate") = TN / (# true non-seizures)
  - Specificity = 1 − False positive rate
- Wikipedia article

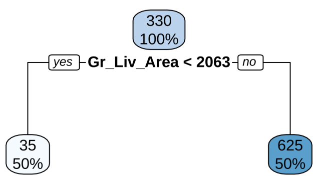
If you were designing a seizure alert system, would you want sensitivity and specificity to be high or low? What are the trade-offs associated with each decision?

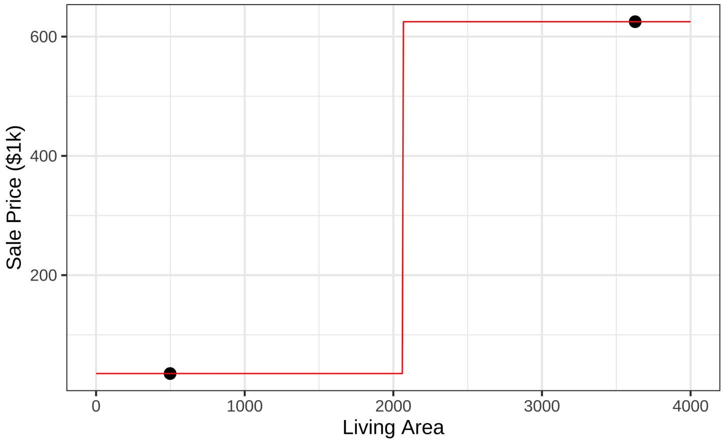
Validation

Key point: you *must* evaluate predictions on *unseen* data

Hey look! I can predict exactly how much a home will sell for!

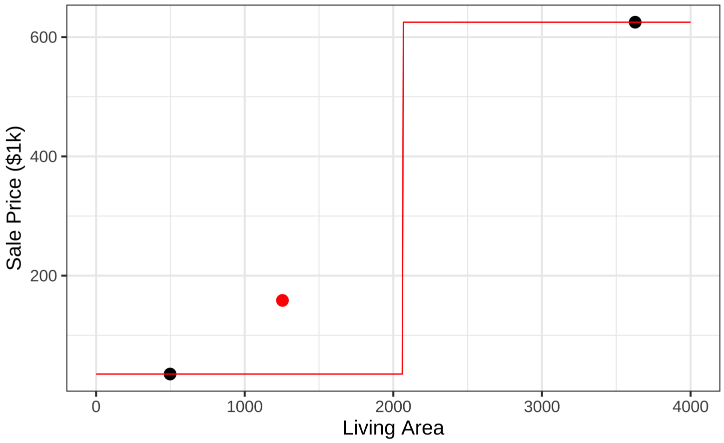
Gr_Liv_Area	Sale_Price
498	35
3627	625





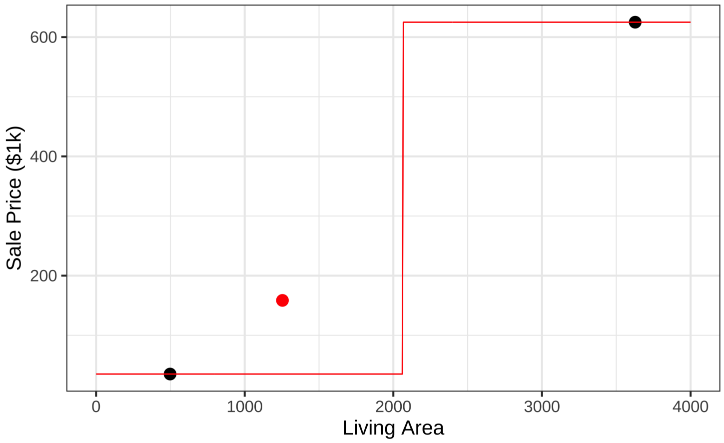
Validation: *unseen* data

Gr_Liv_Area	Sale_Price
498	35.0
3627	625.0
1254	158.5



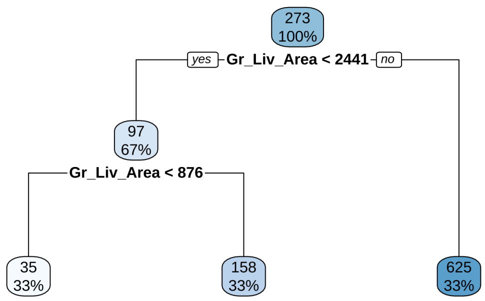
Validation: *unseen* data

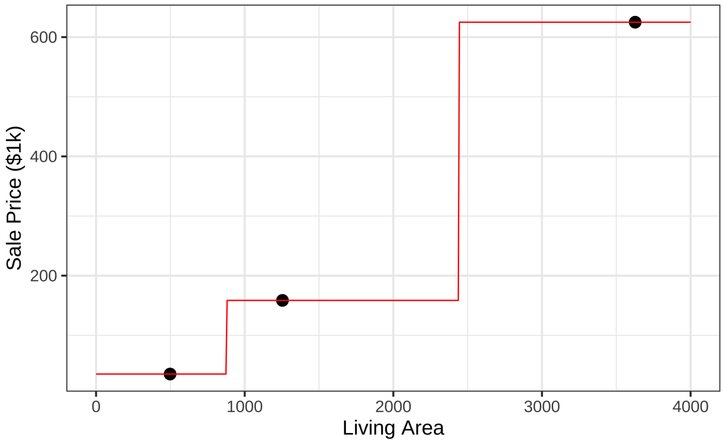
Gr_Liv_Area	Sale_Price
498	35.0
3627	625.0
1254	158.5



Gr_Liv_Area	Sale_Price	.pred	.resid
498	35.0	35	0.0
3627	625.0	625	0.0
1254	158.5	35	123.5

Oh ok, I'll just fix that one...



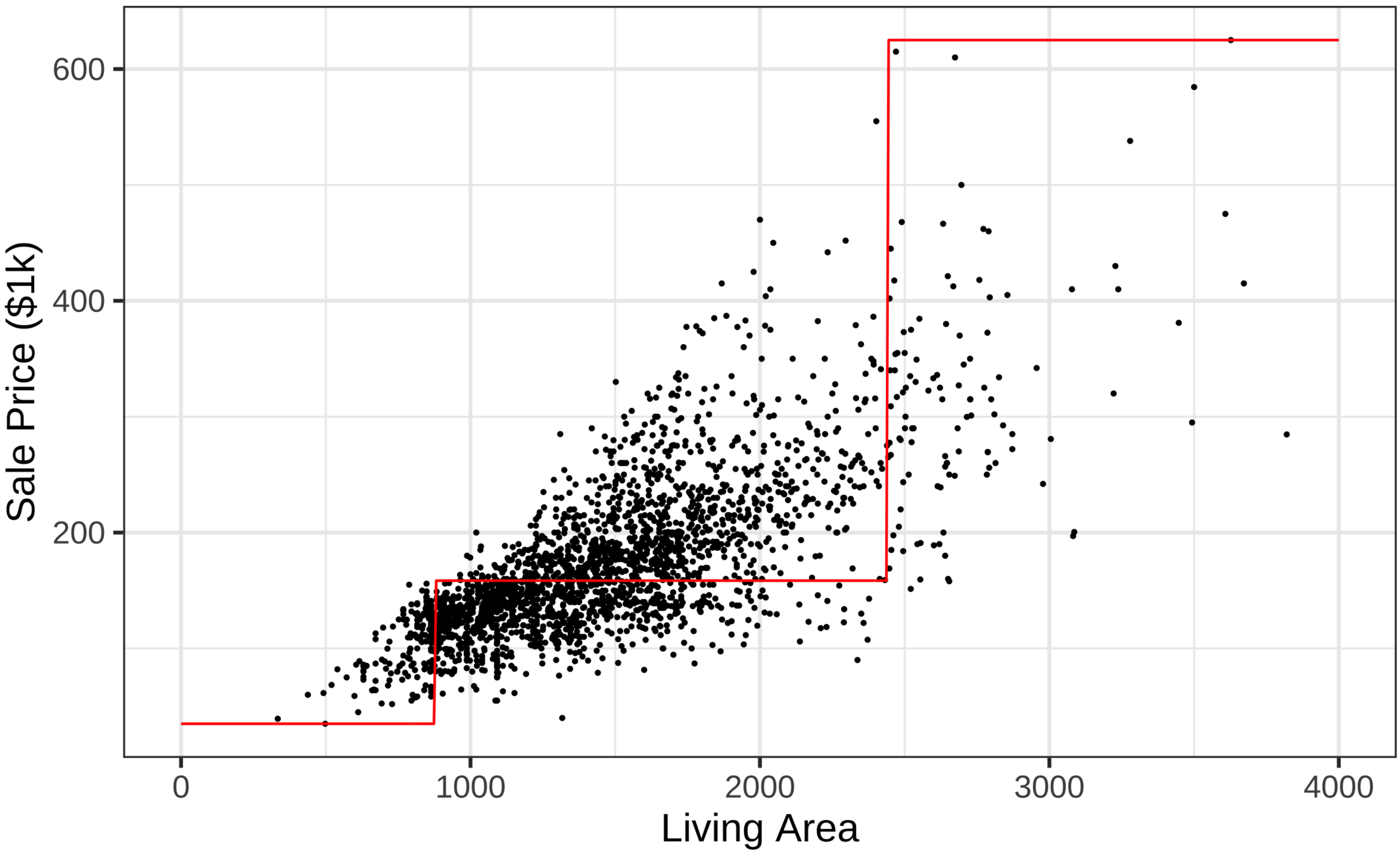


and look, it works!

Gr_Liv_Area	Sale_Price	.pred	.resid
498	35.0	35.0	0
3627	625.0	625.0	0
1254	158.5	158.5	0

Do you really think so?

Now, all the data.



Our model had looked perfect. Where did we go wrong?

Failure to generalize

Predictive models almost always do better on the data they're trained on than anything else.

> Why?

Failure to generalize

Predictive models almost always do better on the data they're trained on than anything else.

> Why?

- model uses a pattern that only held by chance
- model uses a pattern that only holds for some data
- model uses a pattern that's real but got a fuzzy picture of it

How can we accurately assess our models?

> How can we accurately assess our models?

General strategy: *hold out data*.

```
set.seed(10) # Make consistent train-test splits.
ames_split <- initial_split(ames, prop = 2/3)
ames_train <- training(ames_split)
ames_test <- testing(ames_split)
```

2412 total homes:

- 1608 in `ames_train`
- 804 in `ames_test`

> How should we use `ames_train`? `ames_test`?