

STA 235H - Model Selection I: Bias vs Variance, Cross-Validation, and Stepwise

Fall 2022

McCombs School of Business, UT Austin

Last class

- Finished with causal inference, discussing **regression discontinuity designs**
 - We will review the **JITT** (slides will be posted tomorrow)
 - Importance of **doing the coding exercises**

Introduction to prediction

- So far, we had been focusing on **causal inference**:
 - Estimating an effect and "predicting" a counterfactual (what if?)
- Now, we will focus on **prediction**:
 - Estimate/predict outcomes under specific conditions.



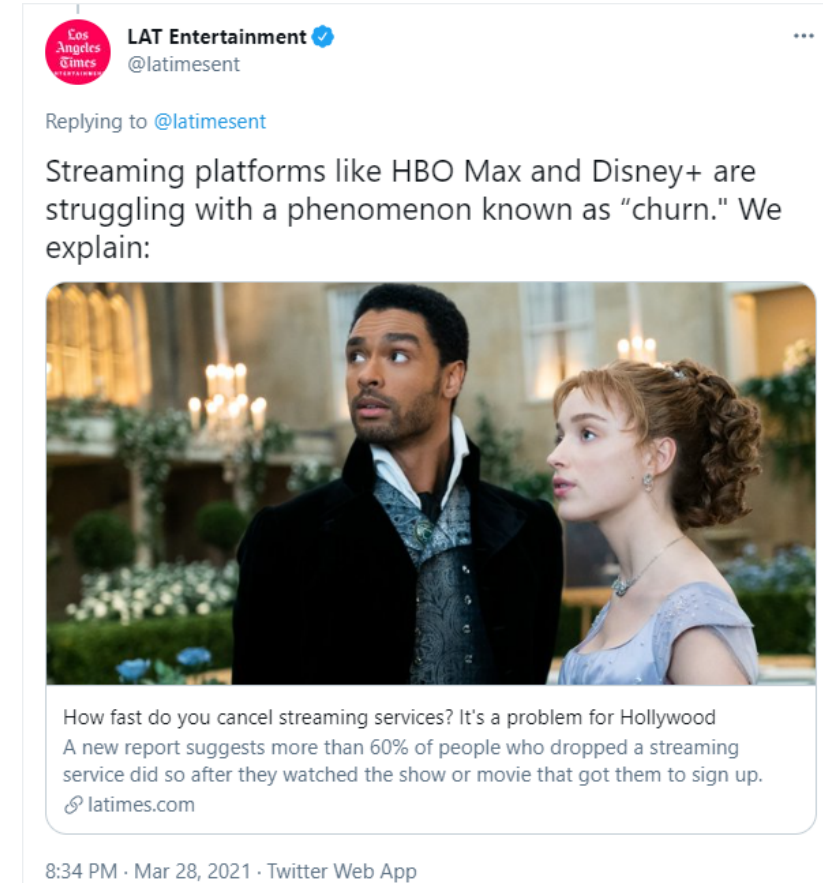
Differences between inference and prediction

- Inference → focus on **covariate**
 - **Interpretability** of model.
- Prediction → focus on **outcome variable**
 - **Accuracy** of model.

Both can be complementary!

Example: What is churn?

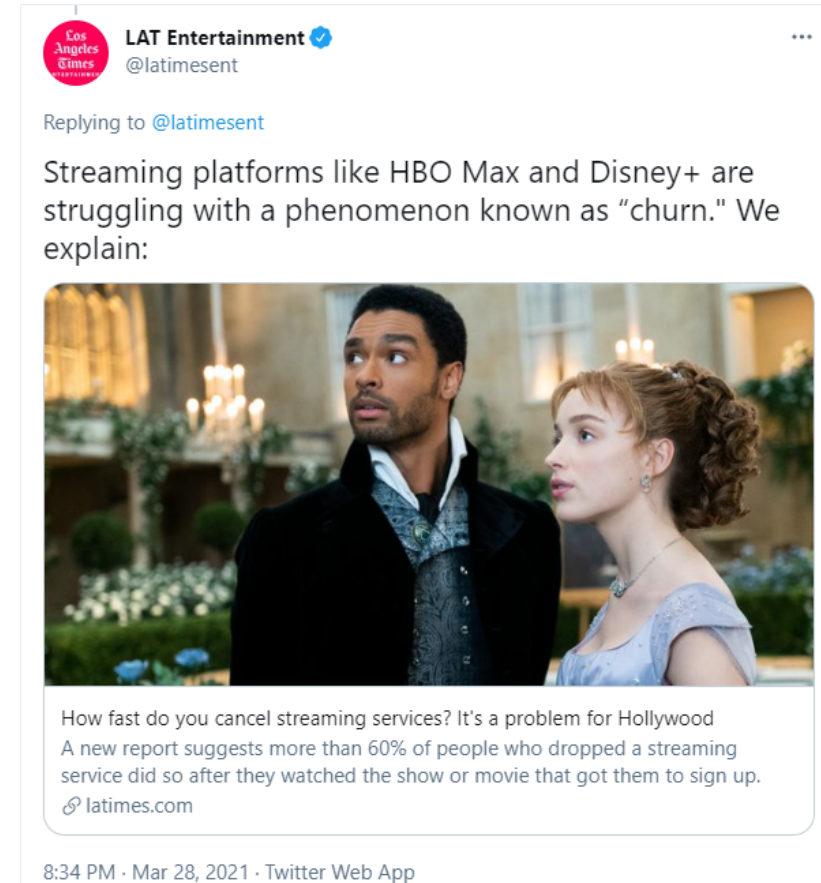
- **Churn:** Measure of how many customers stop using your product (e.g. cancel a subscription).



Example: What is churn?

- **Churn:** Measure of how many customers stop using your product (e.g. cancel a subscription).

Less costly to keep a customer than bring a new one



Example: What is churn?

- **Churn:** Measure of how many customers stop using your product (e.g. cancel a subscription).

Less costly to keep a customer than bring a new one

Prevent churn



Example: What is churn?

- **Churn:** Measure of how many customers stop using your product (e.g. cancel a subscription).

Less costly to keep a customer than bring a new one

Prevent churn

Identify customer that are likely to cancel/quit/fail to renew



Bias vs Variance

"There are no free lunches in statistics"

- Not one method dominates others: Context/dataset dependent.
- Remember that the goal of prediction is to have a method that is accurate in predicting outcomes on **previously unseen data**.
 - **Validation set approach:** Training and testing data

Balance between flexibility and accuracy

Bias vs Variance

Variance

"[T]he amount by which the function f would change if we estimated it using a different training dataset"

Bias

"[E]rror introduced by approximating a real-life problem with a model"

Q1: Which models do you think are higher variance?

a) More flexible models

b) Less flexible models

Bias vs. Variance: The ultimate battle

- In inference, **bias >> variance**
- In prediction, we care about **both**:
 - Measures of accuracy will have both bias and variance.

Trade-off at different rates

How do we measure accuracy?

Different measures:

- Remember $Adj - R^2$?
 - R^2 (proportion of the variation in Y explained by X s) adjusted by the number of predictors!
- **Mean Squared Error (MSE)**: *Can be decomposed into variance and bias terms*

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2$$

- Other measures: Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC)

Is flexibility always better?

Is flexibility always better?

Is flexibility always better?

Is flexibility always better?

Example: Let's predict pre-churn!

- You work at HBO Max and you know that a good measure for someone at risk of unsubscribing is the **times they've logged in the past week**:

```
hbo <- read.csv("https://raw.githubusercontent.com/maibennett/sta235/main/exampleSite/content/Classo
head(hbo)
```

##	id	female	city	age	logins	got	unsubscribe
## 1	1	1	1	53	10	0	1
## 2	2	1	1	48	7	1	0
## 3	3	0	1	45	7	1	0
## 4	4	1	1	51	5	1	0
## 5	5	1	1	45	10	0	0
## 6	6	1	0	40	0	1	0

Two candidates: Simple vs Complex

- Simple Model:

$$\mathit{logins} = \beta_0 + \beta_1 \times \mathit{GoT} + \beta_2 \times \mathit{city} + \varepsilon$$

- Complex Model:

$$\mathit{logins} = \beta_0 + \beta_1 \times \mathit{GoT} + \beta_2 \times \mathit{age} + \beta_3 \times \mathit{age}^2 + \\ \beta_4 \times \mathit{city} + \beta_5 \times \mathit{female} + \varepsilon$$

Create Validation Sets

```
set.seed(100) #Always set seed for replication!  
n <- nrow(hbo)  
train <- sample(1:n, n*0.8) #randomly select 80% of the rows for our training sample  
train.data <- hbo %>% slice(train)  
test.data <- hbo %>% slice(-train)
```

Create Validation Sets

```
set.seed(100) #Always set seed for replication!
```

```
n <- nrow(hbo)
```

```
train <- sample(1:n, n*0.8)
```

```
train.data <- hbo %>% slice(train)
```

```
test.data <- hbo %>% slice(-train)
```

Create Validation Sets

```
set.seed(100) #Always set seed for replication!  
n <- nrow(hbo)  
train <- sample(1:n, n*0.8) #randomly select 80% of the rows for our training sample  
train.data <- hbo %>% slice(train)  
test.data <- hbo %>% slice(-train)
```

Estimate Accuracy Measure

```
library(modelr)

lm_simple <- lm(logins ~ got + city, data = train.data)

lm_complex <- lm(logins ~ female + city + age + I(age^2) + got, data = train.data)

# For simple model:
rmse(lm_simple, test.data)
```

```
## [1] 2.089868
```

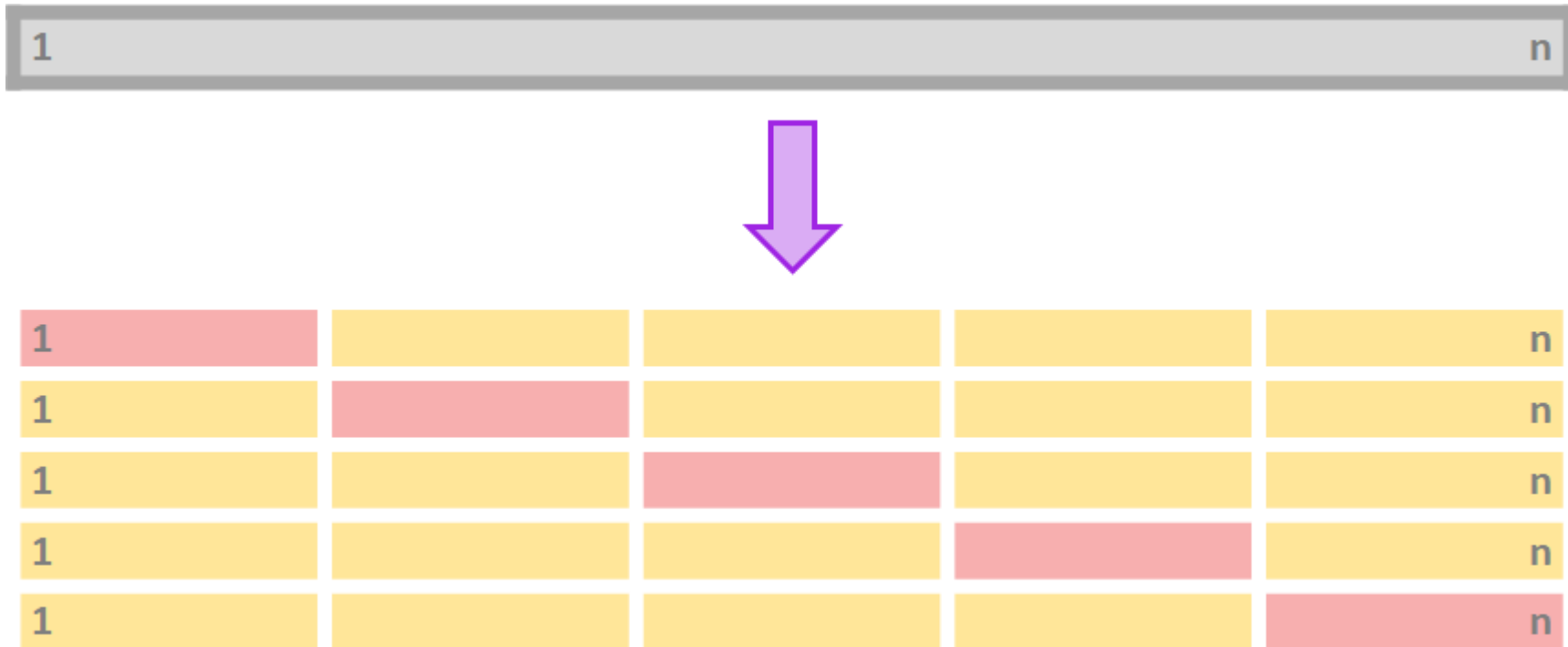
```
# For complex model:
rmse(lm_complex, test.data)
```

```
## [1] 2.093429
```

- Q2: Which one would you prefer?

Cross-Validation

- To avoid using only **one training and testing dataset**, we can iterate over *k-fold* division of our data:



Cross-Validation

Procedure for *k-fold* cross-validation:

1. Divide your data in *k-folds* (usually, $K = 5$ or $K = 10$).
2. Use $k = 1$ as the testing data and $k = 2, \dots, K$ as the training data.
3. Calculate the accuracy measure A_k on the testing data.
4. Repeat for each k .
5. Average A_k for all $k \in K$.

Main advantage: Use the entire dataset for training **AND** testing.

How do we do CV in R?

```
library(caret)

set.seed(100)

train.control <- trainControl(method = "cv", number = 10)

lm_simple <- train(logins ~ got + city, data = disney, method="lm", trControl = train.control)

lm_simple
```

How do we do CV in R?

```
library(caret)

set.seed(100)

train.control <- trainControl(method = "cv", number = 10)

lm_simple <- train(logins ~ got + city, data = disney, method="lm", trControl = train.control)

lm_simple
```

How do we do CV in R?

```
library(caret)
set.seed(100)
train.control <- trainControl(method = "cv", number = 10)
lm_simple <- train(logins ~ got + city, data = disney, method="lm", trControl = train.control)
lm_simple
```

How do we do CV in R?

```
library(caret)

set.seed(100)

train.control <- trainControl(method = "cv", number = 10)

lm_simple <- train(logins ~ got + city, data = hbo, method="lm", trControl = train.control)

lm_simple
```

```
## Linear Regression
##
## 5000 samples
##    2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4500, 4501, 4499, 4500, 4500, 4501, ...
## Resampling results:
##
##    RMSE      Rsquared    MAE
##  2.087314  0.6724741  1.639618
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Stepwise selection

- We have seen how to choose between some given models. **But what if we want to test all possible models?**
- **Stepwise selection:** Computationally-efficient algorithm to select a model based on the data we have (subset selection).

Algorithm for forward stepwise selection:

1. Start with the *null model*, M_0 (no predictors)
2. For $k = 0, \dots, p - 1$: (a) Consider all $p - k$ models that augment M_k with one additional predictor. (b) Choose the *best* among these $p - k$ models and call it M_{k+1} .
3. Select the single best model from M_0, \dots, M_p using CV.

Backwards stepwise follows the same procedure, but starts with the full model.

**Will forward stepwise subsetting
yield the same results as
backwards stepwise selection?**

How do we do stepwise selection in R?

```
set.seed(100)

train.control <- trainControl(method = "cv", number = 10) #set up a 10-fold cv

lm.fwd <- train(logins ~ . - unsubscribe, data = hbo, method = "leapForward",
               tuneGrid = data.frame(nvmax = 1:5), trControl = train.control)

lm.fwd$results
```

##	nvmax	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
## 1	1	2.273269	0.6113410	1.847755	0.04250683	0.01688996	0.04353289
## 2	2	2.087314	0.6724741	1.639618	0.04920703	0.01434646	0.04889721
## 3	3	2.087994	0.6722625	1.640315	0.04919353	0.01436182	0.04904907
## 4	4	2.088156	0.6722088	1.640489	0.04919301	0.01435653	0.04904416
## 5	5	2.088235	0.6721845	1.640525	0.04925197	0.01438207	0.04908729

- Which one would you choose out of the 5 models? Why?

How do we do stepwise selection in R?

```
# We can see the number of covariates that is optimal to choose:  
lm.fwd$bestTune
```

```
##      nvmax  
## 2         2
```

```
# And how does that model looks like:  
summary(lm.fwd$finalModel)
```

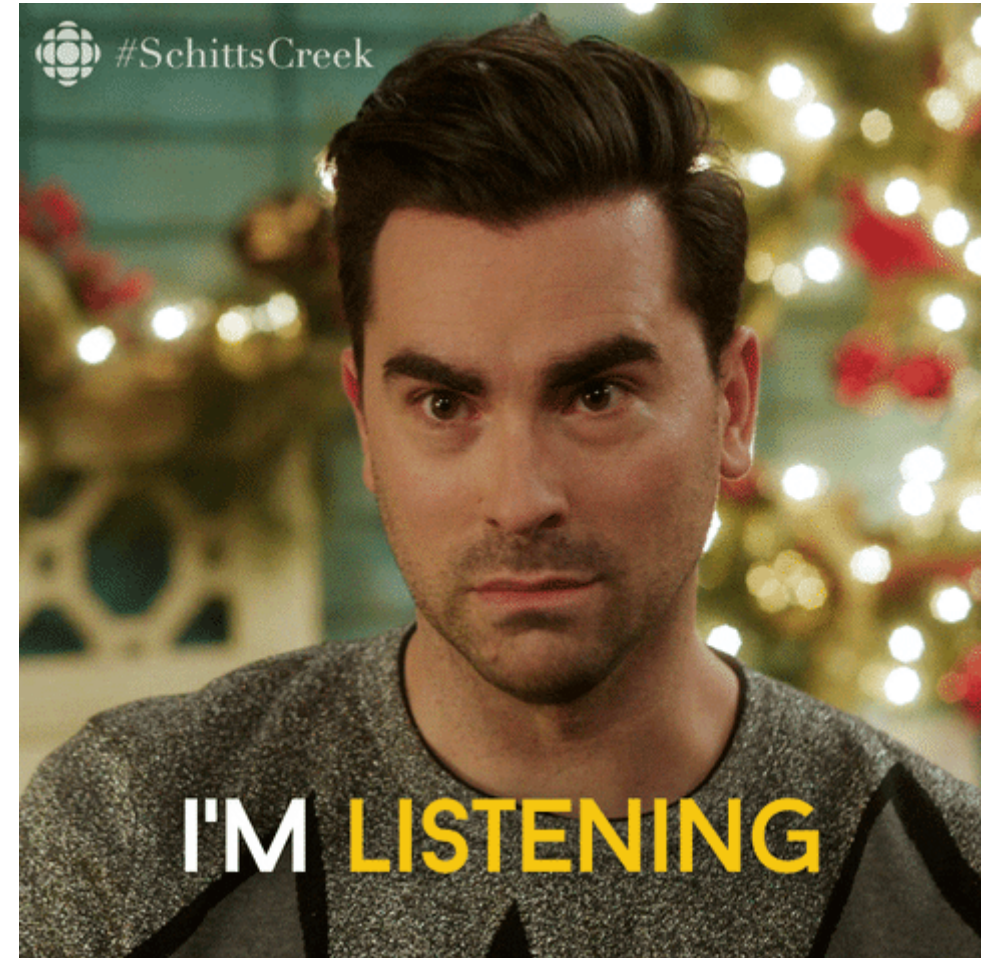
```
## Subset selection object  
## 5 Variables (and intercept)  
##      Forced in Forced out  
## id      FALSE      FALSE  
## female  FALSE      FALSE  
## city     FALSE      FALSE  
## age      FALSE      FALSE  
## got      FALSE      FALSE  
## 1 subsets of each size up to 2  
## Selection Algorithm: forward  
##      id female city age got  
## 1 ( 1 ) " " " " " " " "*"  
## 2 ( 1 ) " " " " " "* " " "*"
```

```
# If we want to recover the coefficient names, we can use the coef() function:  
coef(lm.fwd$finalModel, lm.fwd$bestTune$nvmax)
```

```
## (Intercept)      city      got  
##    7.026494    2.577163   -6.265728
```

Takeaway points

- In prediction, everything is going to be about **bias vs variance**.
- Importance of **validation sets**.
- We now have methods to **select models**.



Next class

- Continue with prediction and model selection
- **Shrinkage/Regularization methods:**
 - Ridge regression and Lasso.



References

- James, G. et al. (2021). "Introduction to Statistical Learning with Applications in R". *Springer. Chapter 2, 5, and 6.*
- STDHA. (2018). "Stepwise Regression Essentials in R."
- STDHA. (2018). "Cross-Validation Essentials in R."