# Beyond Just Prediction

## Causal Thinking in Machine Learning

**Avik Basu**

**(He/Him)**

# About Me

- Based in Sunnyvale, CA
- Staff Data Scientist at Intuit
- Editor at PyOpenSci
- Engineering + Data Science
- Love PS5 + Steam Deck 🎮
- Football + Tennis 🎾
- Driving is therapy! 🛞

# Agenda

1. Correlation vs Causation
2. Causal Concepts
3. Uplift Modeling
4. Evaluation
5. Key Takeaways

# The Prediction problem

## Customer Churn

**Scenario**: Build a model to predict customer churn

**Performance**: F1 score of 0.9

**Model learned**:
*P(Churn | Called Support) = 0.45* vs *P(Churn | No Call) = 0.15*

**Model insight**: Customers who call support are 3x more likely to churn

# The Prediction problem

## Customer Churn

**Business action:** “How about we reduce support calls”?

**Confusion:** Did support calls *cause* churn, or did unhappy customers call support?

**Key Insight:** “High Accuracy != Actionable Insights”

# Correlation vs Causation

## Prediction vs Intervention

| | Predictive Models | Causal Models |
|---|---|---|
| **Estimates** | $E[Y \mid X]$ | $E[Y^1 - Y^0 \mid X]$ |
| **Question** | Who will buy? | Who will buy because of the treatment? |
| **Answers** | What will happen? | What is the effect of our action? |

# Correlation vs Causation

## Example

**Problem**

You run an email marketing campaign and want to measure its effectiveness.

**Prediction**

- E[Purchase | Received Email] = 0.2
- 20% of customers who got email purchased

**Intervention**

- E[Purchase | T=1, X] - E[Purchase | T=0, X] = 0.05
- T = 1 (== “Email Sent”)
- Sending email increases purchase rate by 5%

# The Selection Bias Problem

## Why we can’t just compare groups

| Group | Purchase Rate |
| --- | --- |
| Received Email (Treatment) | 20% |
| No Email (Control) | 8% |

**Conclusion**

Email boosts purchases by 12%!!!

# The Selection Bias Problem

**But wait.. who got the emails?**

**Email Recipients**

- Recently active customers
- High engagement history

**No Email**

- Inactive customers
- Low engagement history

**<u>Problem</u>**

- Engaged customers are more likely to purchase regardless of email
- Observed 12% difference = True effect + Selection Bias

# The Selection Bias Problem

## The Real Picture

| Customer Type | Email Group | No-Email Group |
|---|---|---|
| High Engagement | 20% purchase | 16% purchase |
| Low Engagement | 6% purchase | 3% purchase |

- True effect for High-engagement: 20% - 16% = 4%
- True effect for Low-engagement: 6% - 3% = 3%

# Randomized A/B Tests

## Solve Selection Bias!

**Key Idea**

Random assignments ensures treatment and control groups are identical at the baseline.

**After Randomization**

Treatment group (Email):  Random 50% of ALL customers

Control group (No Email):  Random 50% of ALL customers

Groups identical on average → Observed difference = True effect

# **Problem?**

A/B tests are *expensive* and sometimes *unethical*.

# How do we estimate treatment effects from observational data?

# Uplift Modeling

# Uplift Modeling

## 3 Key components

1. **Potential Outcomes**
   How to think about treatment effects

2. **CATE**
   What we're trying to estimate

3. **Meta-Learners**
   How we estimate CATE

# Potential Outcomes Framework

## How to think about treatment effects

- For each individual there are 2 potential outcomes
    - $Y_i^0$ = outcome for person i if NOT treated (control)
    - $Y_i^1$ = outcome for person i if treated (treatment)

- **Individual Treatment Effect** (ITE) = $Y_i^1 - Y_i^0$

# Potential Outcomes Framework

## Customer example

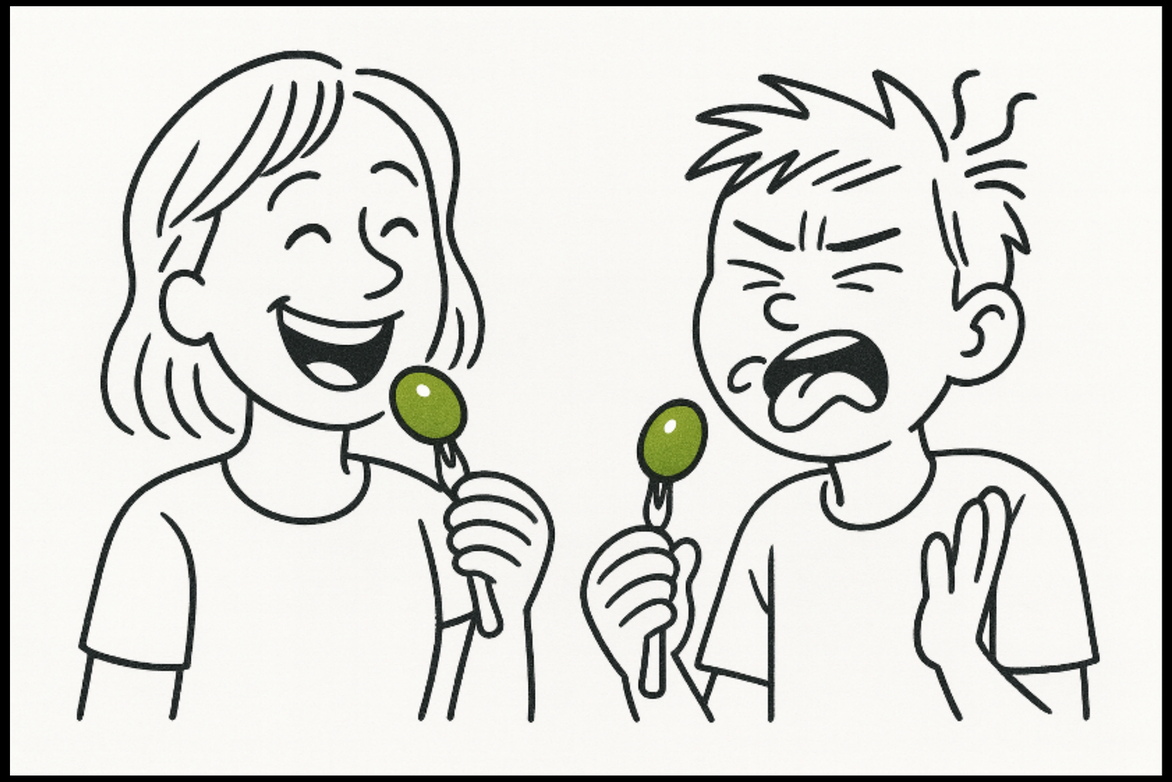
- Maria got an email (T=1) and did a purchase (Y = 1)
- We observe: $Y^1 = 1$
- We DO NOT observe $Y^0 = ?$ (Counterfactual)
- Would she have purchased WITHOUT the email?
  - ★ If $Y^0 = 1$ => ITE = 0 (would have bought it anyway)
  - ★ If $Y^0 = 0$ => ITE = 1 (email caused the purchase)

# Uplift Modeling

## Motivation

- Goal is to provide intervention to the right customers

- Individuals respond differently to interventions which gets hidden in the average



Same treatment, different reactions!

# 4 types of people

## Hypothetical scenario

1 = Purchased

0 = Not Purchased

| Customer | Received Email ($Y^1$) | No Email ($Y^0$) | ITE ($Y^1 - Y^0$) | Customer Type |
|---|---|---|---|---|
| **Mr. Grayson** | 1 | 1 | 0 | Sure Things! 👍🏼 |
| **Mr. Drake** | 1 | 0 | 1 | Persuadables 💸 |
| **Ms. Gordon** | 0 | 0 | 0 | Lost causes 👎🏼 |
| **Ms. Kyle** | 0 | 1 | -1 | Sleeping dogs 🐶 |

ATE = 0

# 4 types of people

## Hypothetical scenario

1 = Purchased

0 = Not Purchased

| Customer | Received Email ($Y^1$) | No Email ($Y^0$) | ITE ($Y^1 - Y^0$) | Customer Type |
|---|---|---|---|---|
| **Mr. Grayson** | 1 | 1 | 0 | Sure Things! 👍🏼 |
| **Mr. Drake** | 1 | 0 | 1 | Persuadables 💸 |
| **Ms. Kyle** | 0 | 1 | -1 | Lost causes 👎🏼 |
| **Ms. Gordon** | 0 | 0 | 0 | Sleeping dogs 🐶 |

ATE = 0

# ITE is unavailable!

## Real scenario

| Customer | Received Email ($Y^1$) | No Email ($Y^0$) | ITE ($Y^1 - Y^0$) |
|---|---|---|---|
| **Mr. Grayson** | 1 | - | ?? |
| **Mr. Drake** | - | 0 | ?? |
| **Ms. Kyle** | - | 1 | ?? |
| **Ms. Gordon** | 0 | - | ?? |
| **Mr. Wayne** | 1 | - | ?? |
| **Ms. Quinn** | - | 1 | ?? |

# Conditional Average Treatment Effects (CATE)

**What uplift models predict!**

**Problem**

We cannot observe ITE for any individual.

**Solution**

- $CATE(x) = E[Y^1 - Y^0 \mid X = x]$
- Expected uplift or treatment effect for customers with similar characteristics/ features

# Uplift Modeling

## Estimating CATE

| Customer | Received Email ($Y^1$) | No Email ($Y^0$) | ITE ($Y^1 - Y^0$) |
|---|---|---|---|
| **Mr. Grayson** | 1 | 0.95 | 0.05 |
| **Mr. Drake** | 0.82 | 0 | 0.82 |
| **Ms. Kyle** | 0.44 | 1 | -0.56 |
| **Ms. Gordon** | 0 | 0.37 | -0.37 |
| **Mr. Wayne** | 1 | 0.75 | 0.25 |
| **Ms. Quinn** | 0.99 | 1 | -0.01 |

These estimates are CATE estimates from our uplift model.

# Uplift Modeling

## Meta Learners

- Way to leverage predictive ML methods for estimating treatment effects
- The underlying predictive model can be any model type
- The predictive model can be estimating regression or classification
- Different types of meta learners
  - **S - Learner**
  - **T - Learner**
  - X - Learner

# S-Learner

## Single Model for everything!

- Train one model on all data with treatment as a feature
- Predict both outcomes and compute uplift

| ID | Engagement ($X_1$) | Tenure ($X_2$) | Is Email Sent (T) | Is Purchased (Y) |
|---|---|---|---|---|
| 1 | 0.9 | 5.2 | 1 | 1 |
| 2 | 0.3 | 1.5 | 1 | 0 |
| 3 | 0.8 | 4.7 | 0 | 1 |
| 4 | 0.2 | 0.6 | 0 | 0 |

✓ Simplest technique, and works with limited data really well

✳ Treatment is just another feature, so might not capture complex interactions

# S-Learner

## Logic

```
# 1. Combine feature and treatment
X_combined = [features + treatment]

# 2. Train model
model = LGBMClassifier()
model.fit(X_combined, y)

# 3. Predict both outcomes for a new user with features x
y_1 = model.predict([x, T=1])
y_0 = model.predict([x, T=0])

# 4. Compute uplift (CATE)
tau = y_1 - y_0
```

| ID | ($X_1$) | ($X_2$) | (T) | (Y) |
|---|---|---|---|---|
| **1** | 0.9 | 5.2 | 1 | 1 |
| **2** | 0.3 | 1.5 | 1 | 0 |
| **3** | 0.8 | 4.7 | 0 | 1 |
| **4** | 0.2 | 0.6 | 0 | 0 |

# T-Learner

## Two (at least) model approach

**Model 1**: Trained on Email Recipients (T=1)

| ID | Engagement ($X_1$) | Tenure ($X_2$) | Is Purchased (Y) |
|---|---|---|---|
| **1** | 0.9 | 5.2 | 1 |
| **2** | 0.3 | 1.5 | 0 |

**Model 0**: Trained on No Email (T=0)

| ID | Engagement ($X_1$) | Tenure ($X_2$) | Is Purchased (Y) |
|---|---|---|---|
| **3** | 0.8 | 4.7 | 1 |
| **4** | 0.2 | 0.6 | 0 |

- Model can understand different patterns in treatment vs control
- Better capture heterogeneous treatment effects
- Can tune the 2 models differently
- Needs sufficient data in both groups

# T-Learner

## Logic

```
# 1. Split data by treatment
X_treated, X_control, y_treated, y_control = split_by_recipe(data)

# 2. Train treatment model
model_1 = LGBMClassifier()
model_1.fit(X_treated, y_treated)

# 3. Train control model
model_0 = RandomForestClassifer()
model_0.fit(X_control, y_control)

# 4. For a new person, predict using both models
y_1 = model_1.predict(X_test)
y_0 = model_0.predict(X_test)

# 5. Compute uplift (CATE)
tau = y_1 - y_0
```

# Evaluation

# Evaluation

## Traditional ML metrics fail for uplift

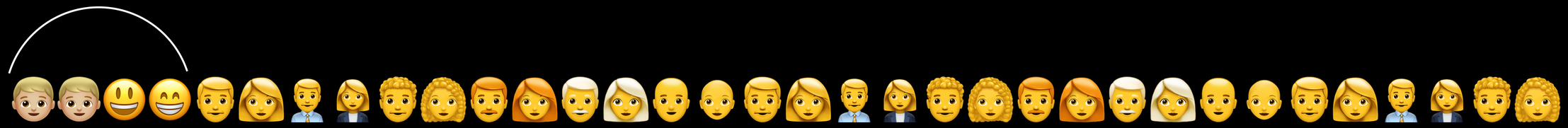
| What it measures | What we need |
| --- | --- |
| Accuracy/F1: How well we predict Y | How well we predict uplift, i.e. $Y^1$ - $Y^0$ |
| “Who will buy?” | “Who will buy BECAUSE of the treatment?” |

# Actual Uplift by Bin

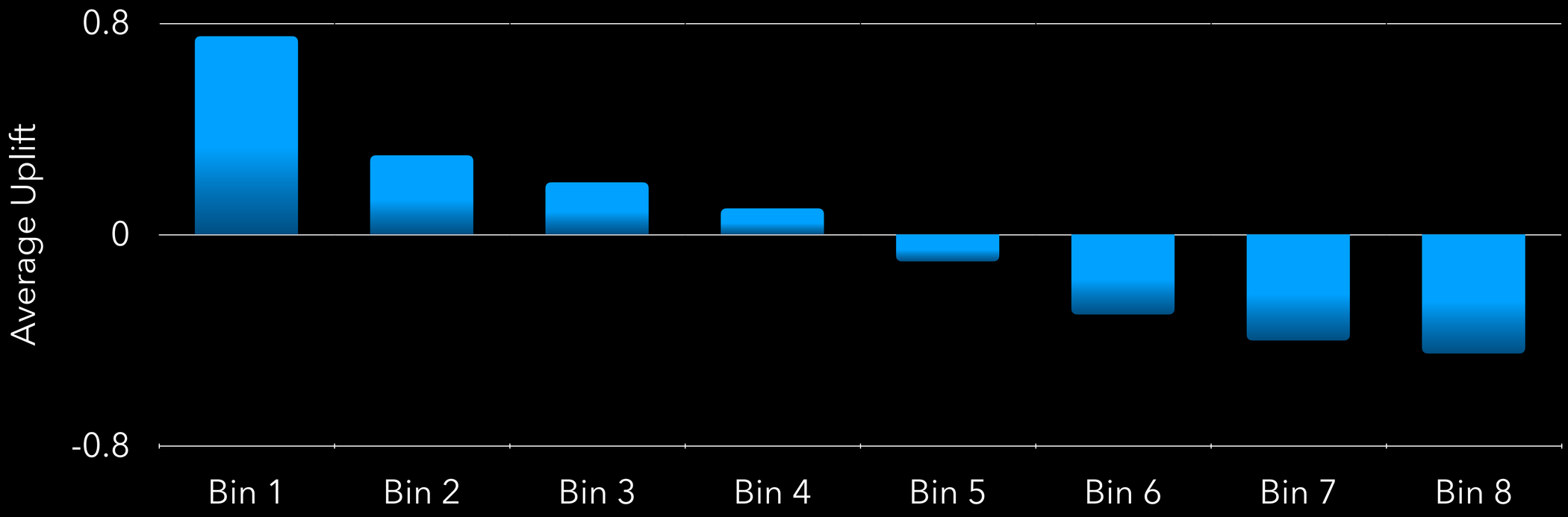
## Evaluation Plot

1. Rank users by predicted uplift score (descending)
2. Divide them into bins (e.g. 10 bins)
3. Calculate $\bar{Y}_b^1 - \bar{Y}_b^0$ to calculate the average uplift in each bucket “b” where,
   - $\bar{Y}_b^1$ is the average outcome in the treatment group and bucket “b”
   - $\bar{Y}_b^0$ is the average outcome in the control group and bucket “b”

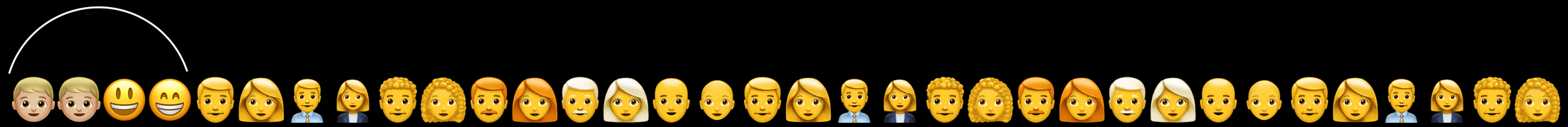
$$\bar{Y}_b^1 - \bar{Y}_b^0$$



# Actual Uplift by Bin



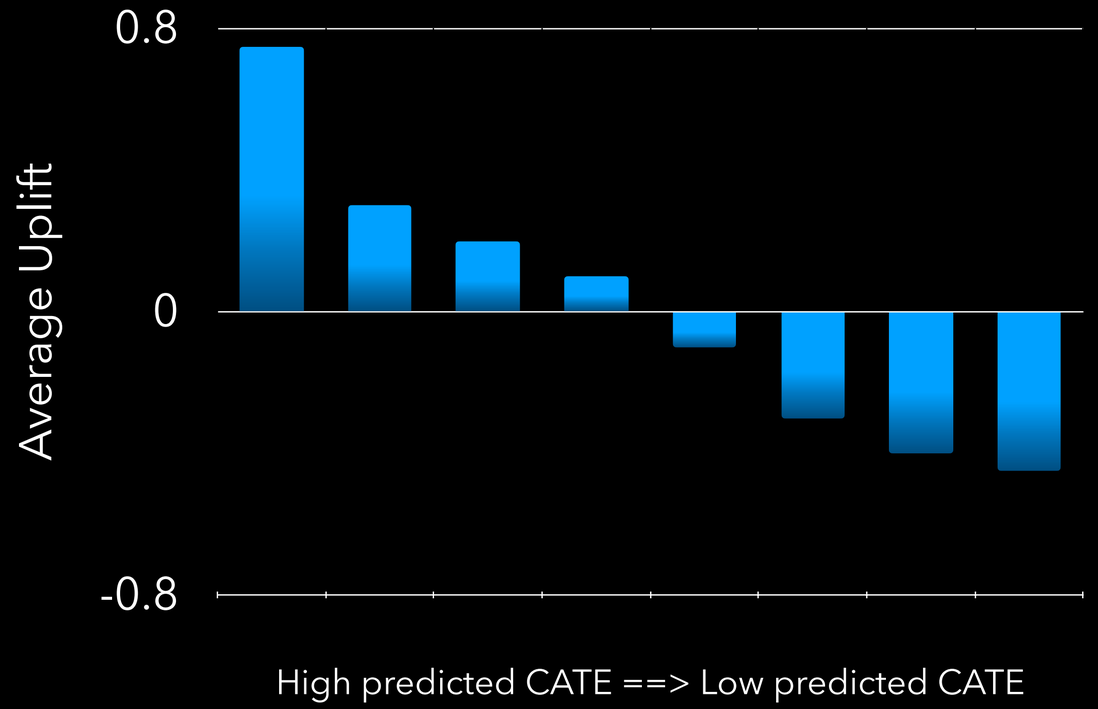

$$\bar{Y}_b^1 - \bar{Y}_b^0$$



# Actual Uplift by Bin

## Takeaways

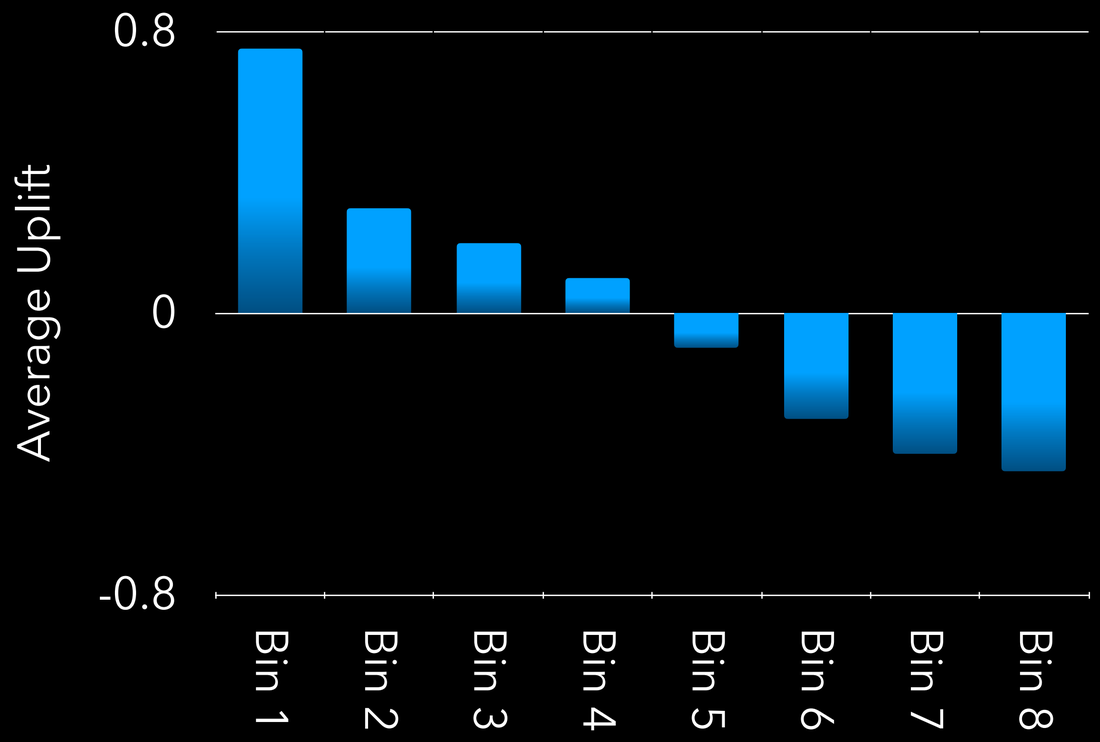
- In the best case we expect a monotonically decreasing pattern as we move from left to right
- If uplift rises or stays flat as the bins increase, it suggests that the model is probably overfitting
- If early bars are close to zero, then the model has a weak uplift
- If less data, then use fewer bins to remove noise



# Customer Targeting Strategy

## Multiple options

1. Target all positive uplift (Bins 1-4)
    - More customers reached
    - Lower average uplift per customer
2. Target only top uplift (Bins 1-2)
    - Fewer customers but highest impact
    - Better resource efficiency



# Key Takeaways

# When to use uplift modeling?

**Use Uplift**

- Can’t treat everyone!
- Treatment has costs (time, money, ethical reasons, customer experience)
- Need to optimize WHO to target

**Stick with Traditional ML**

- Can treat everyone
- Just need predictions; not decisions
- Treatment is virtually free for the business

# Practical Implementation Tips

## Best Practices

- S-Learner is simpler; T-Learner better with sufficient data
- Sufficient samples in both treatment and control data from randomized A/B tests
- Validate with uplift curves (e.g. actual uplift by bin)

## Python Libraries

- Causalml (Uber)
- Scikit-uplift
- econML (Microsoft)

# Thank You! 🙏

Github with code and slides

Connect on LinkedIn