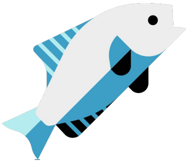
Observational Causal Inference



Customer Value Changing Events: (CeViChE)

Aug 2021

Presenter: Jing Pan[1], Huigang Chen[2]

Uber CausalML

[1] Uber Technologies [2] Facebook

Agenda

01 Motivations
02 CausalML Solution
03 Case Study
04 Summary & Takeaways

KDD website: https://t.uber.com/kdd_causalml
(or https://causal-machine-learning.github.io/kdd2021-tutorial/)
Installation: https://github.com/uber/causalml#installation
Tutorial Notebook: t.uber.com/kdd_causalml_case1

Motivation | Where AB Test Fails

Although true experiments have higher internal validity, sometimes in reality it's **not feasible (not practical or ethical)** to provide or withhold a treatment on a random basis;

Observational Causal Inference can still allow you to establish the causal relationship and estimate the effects (ATE).

CausalML Solution

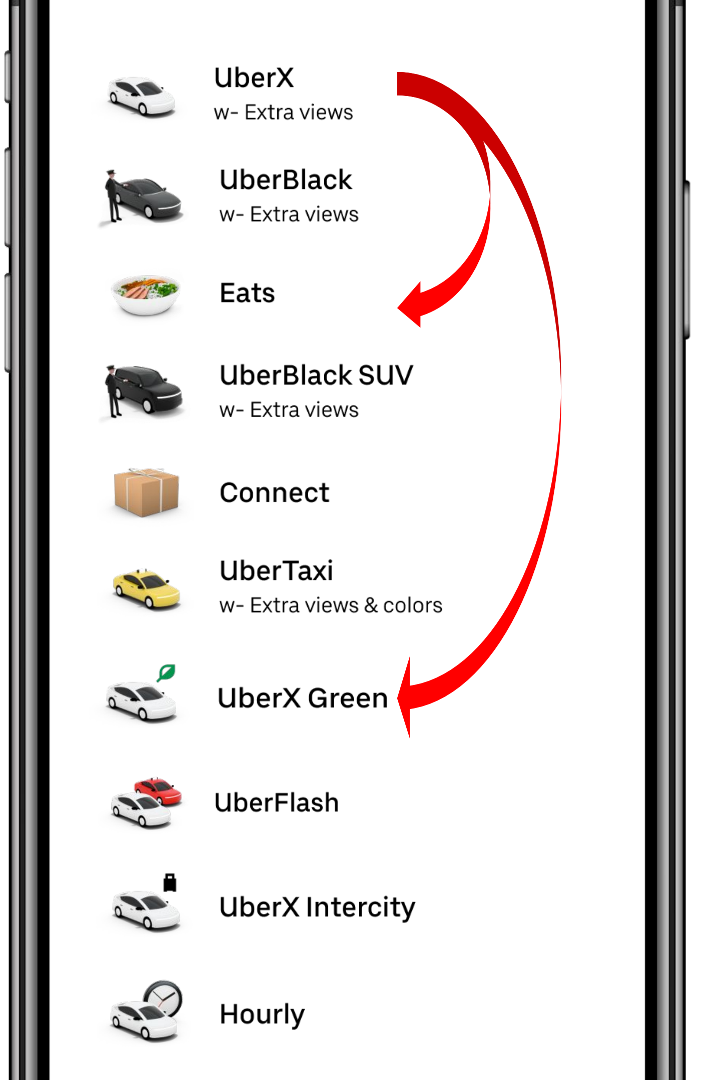
Scoring, Matching, Inference

Motivation

Motivation | Uber

Cross-sell opportunities are huge but require rigor in quantifying the long term monetary impact

Marketers have no way to tie their acquisition efforts to monetary impact for Uber for cross-sell budget planning



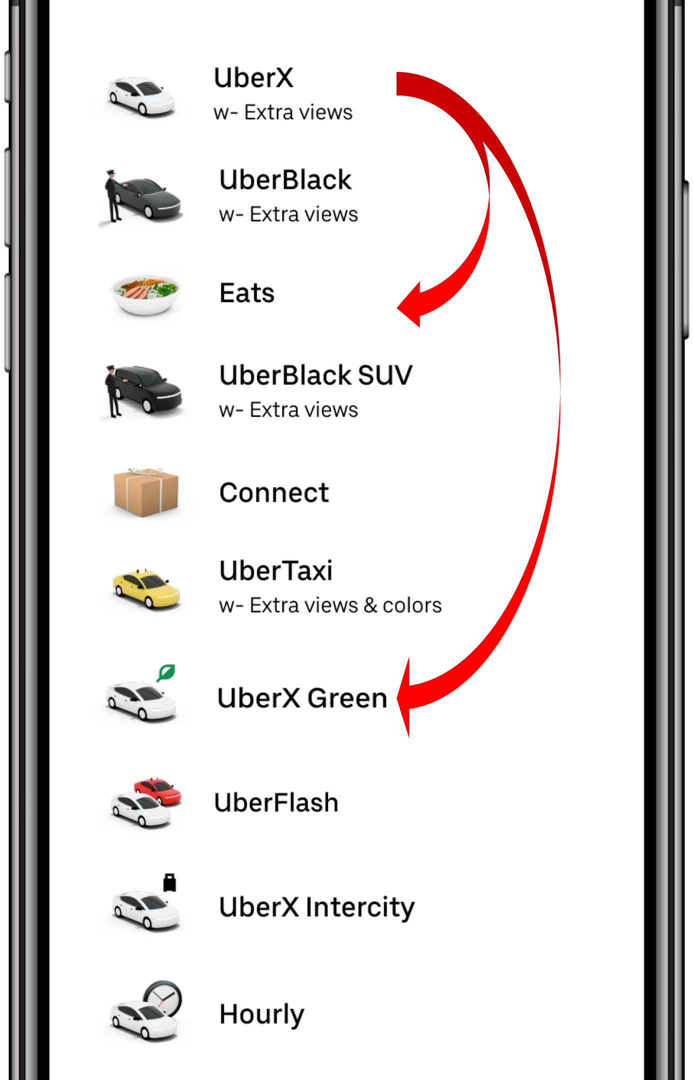
Graph credit: Ben Draves intern at Uber Marketing Data Science team

Motivation | Uber

Cross-sell opportunities are huge but require rigor in quantifying the long term monetary impact

Marketers have no way to tie their acquisition efforts to monetary impact for Uber for cross-sell budget planning

Leadership cannot rely on rigorous causality to prioritize between products acquisition



Graph credit: [Ben Draves]() intern at Uber Marketing Data Science team

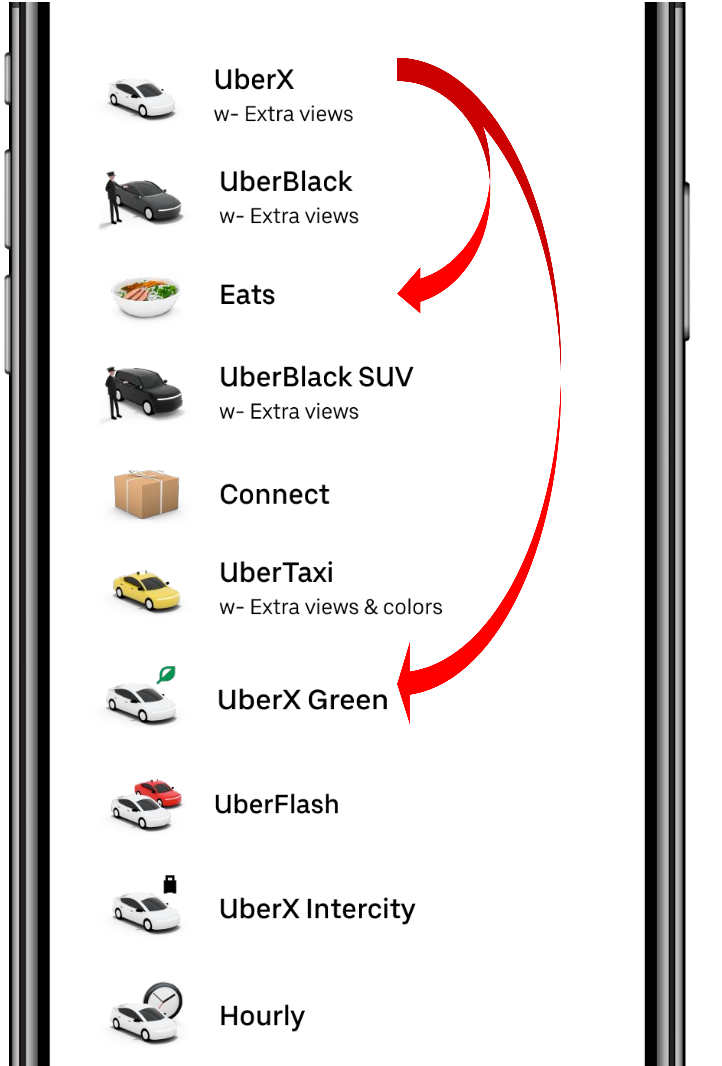
Motivation | Uber

Cross-sell opportunities are huge but require rigor in quantifying the long term monetary impact

Marketers have no way to tie their acquisition efforts to monetary impact for Uber for cross-sell budget planning

Leadership cannot rely on rigorous causality to prioritize between product acquisition

Data Scientists that build cross-sell conversion models cannot measure expected value of a conversion without insight into the value of a conversion on the individual level



Graph credit: Ben Draves intern at Uber Marketing Data Science team

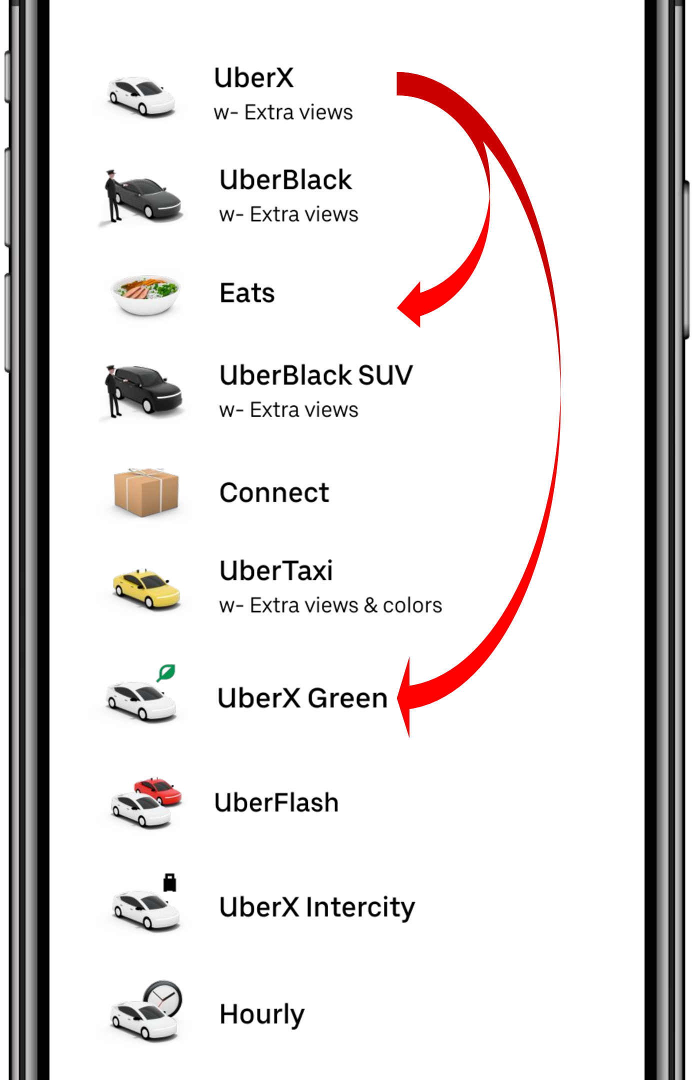
Motivation | Uber

Cross-sell opportunities are huge but require rigor in quantifying the long term monetary impact

Can’t set up a randomized control experiment

Want to understand:

- ❓ **Will promoting cross-sell hurt or help the business at Uber as a whole platform**
- ❓ **How much the incremental impact will be**



Graph credit: [Ben Draves](#) intern at Uber Marketing Data Science team

Motivation | Benefits

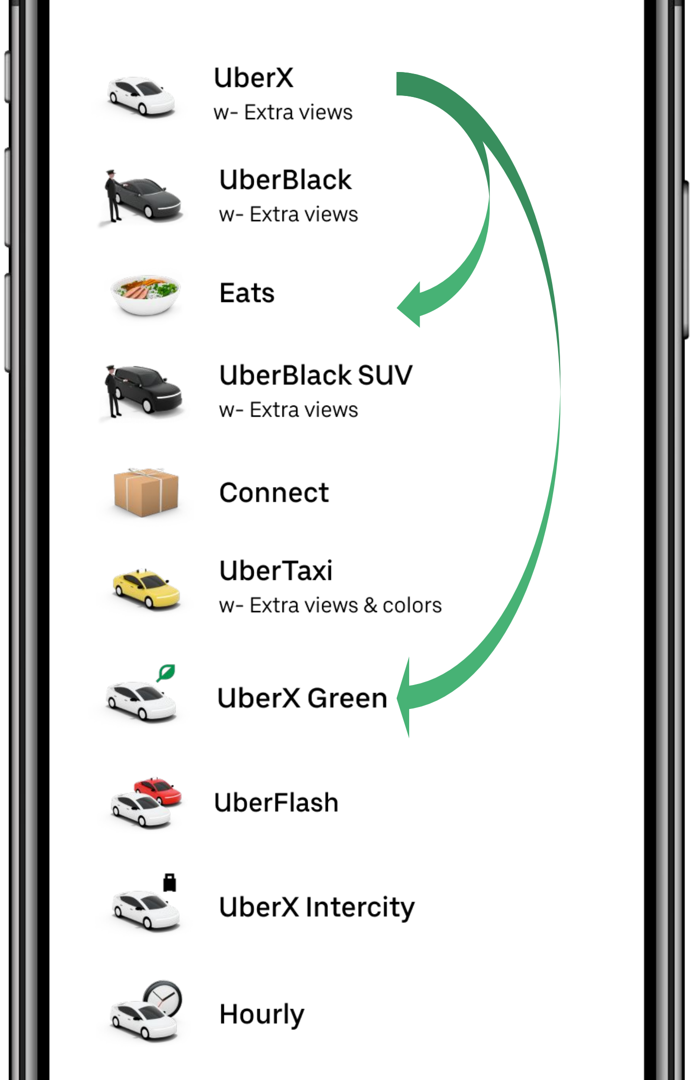
CausalML: leveraging observational data to establish causality

Time horizon relative to field experiments

Not as expensive

Externally valid inference (depending on the data and design)

Less fraught with behavioral concerns



Graph credit: Ben Draves intern at Uber Marketing Data Science team

Motivation | Benefits

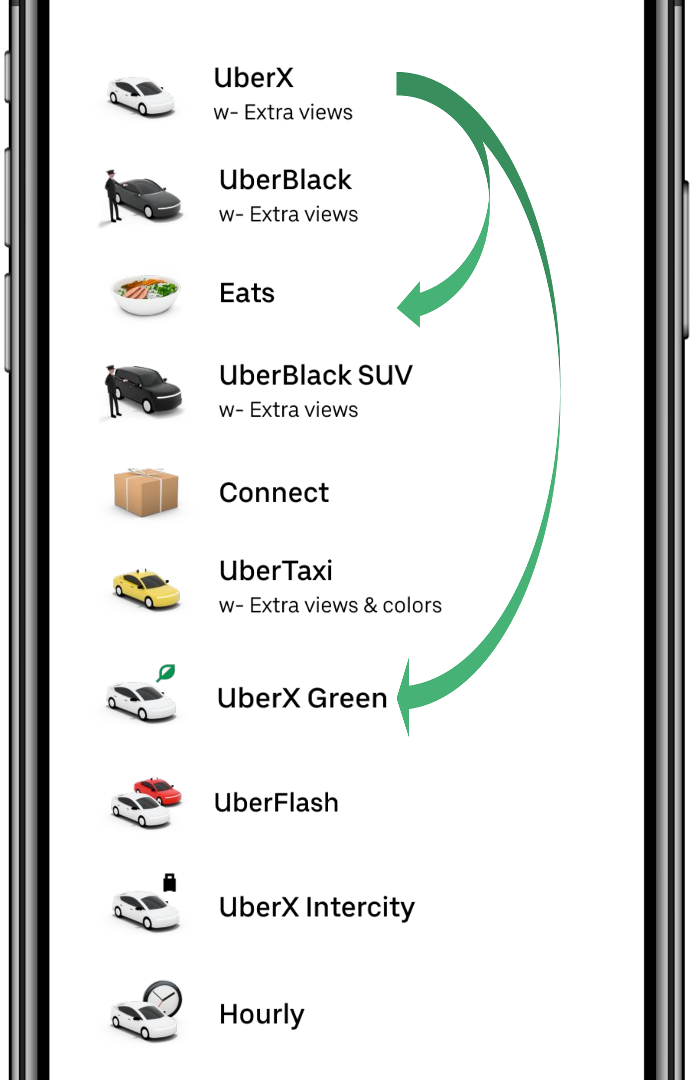
CausalML: leveraging observational data to establish causality

Time horizon relative to field experiments

Not as expensive

Externally valid inference (depending on the data and design)

Less fraught with behavioral concerns



Graph credit: Ben Draves intern at Uber Marketing Data Science team

Motivation | Applications

Generalized the approach applied in multiple scenarios when AB test fails

Cross-sell/Upsell

Disengagement

Loyalty Program

Brand Impact

Marketing Campaigns

...

Note: Data analyzed in this study is observational, not experimental; We applied state-of-the-art causal inference techniques to infer causality. However in practice is never guaranteed, we recommend conducting AB tests to evaluate the implemented strategies.

CeViChE Framework

Validation

Inference

Matching

Propensity Score

Analysis Setup

Methodology

Among all the methods, CeViChE uses well established, tunable matching and rigorous inference validation techniques

- Parametric Regression
  - Difference in Differences (DID pre-post with-without comparison)
  - Regression Discontinuity Design (RDD)
  - Interrupted Time Series
  - Panel Regression
  - …
- Matching
  - Exact Matching
  - Coarsened Exact Matching (CEM)
  - …
- **CeViChE - Stratified Propensity Score Matching + Meta Learners + Sensitivity Analysis**

CausalML	Propensity Score Methods	Causal Inference Models	Sensitivity Analysis

CeViChE Framework

Validation

Inference

Matching

Propensity Score

Analysis Setup

Uber

Uber Eats

CeViChE Framework

Analysis Setup

- **Cohort Preference**
- **Feature Collection***
  - Static Features
  - Time Varying Features
  - LTV Features
  - Marketing
  - Communication
  - Promotion
  - ...
- **Feature Engineering**

* As in any observational study, there may still exist hidden confounders that influence both the conversion behavior and the potential outcomes. We try to be comprehensive in the feature list for the matching to remove the bias as much as we can.

CeViChE Framework

Propensity Score Matching: Homogeneous group creation from observation data

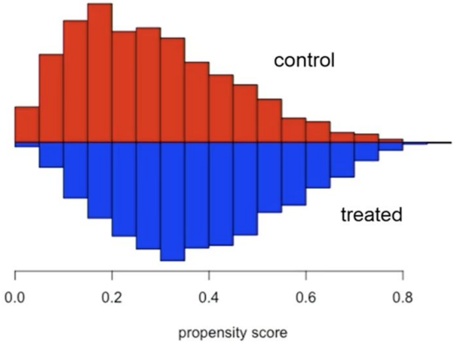
- **Propensity score matching*:** User-level matching procedure to ensure T/C groups are as similar as possible.
  - Propensity score reflects each user's characteristics and past behaviours.
  - Perform a nearest neighbor matching based on propensity score (notable point: add calipers to define the maximum threshold for difference).
  - Optimize the matching results by adding matching covariat together with propensity score into matching.
  - Verify similarity of and feature distributions between treatment and control after matching.

```python
from causalml.match import NearestNeighborMatch, create_table_one, MatchOptimizer
from causalml.propensity import ElasticNetPropensityModel


pm = ElasticNetPropensityModel(random_state=RANDOM_SEED)
pm.fit(df_train[PROPENSITY_FEATURES], df_train[TREATMENT_COL])
df['pihat']  = pm.predict(df[PROPENSITY_FEATURES])

matcher = NearestNeighborMatch(caliper=0.1, replace=True)
df_matched = matcher.match(data=df, treatment_col=TREATMENT_COL, score_cols=SCORE_COL)

optimizer = MatchOptimizer(treatment_col=TREATMENT_COL, ps_col=SCORE_COL,
                           matching_covariates=[SCORE_COL],
                           min_users_per_group=100, smd_cols=[SCORE_COL])
df_matched_opt = optimizer.search_best_match(df)
```

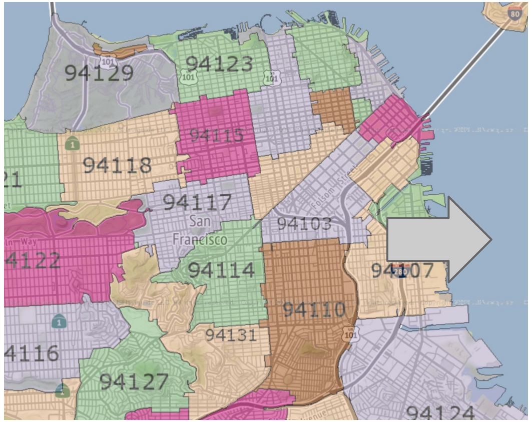
* As in any observational study, there may still exist hidden confounders that influence both the conversion behavior and the potential outcomes. We try to be comprehensive in the feature list for the matching to remove the bias as much as we can.

CeViChE Framework

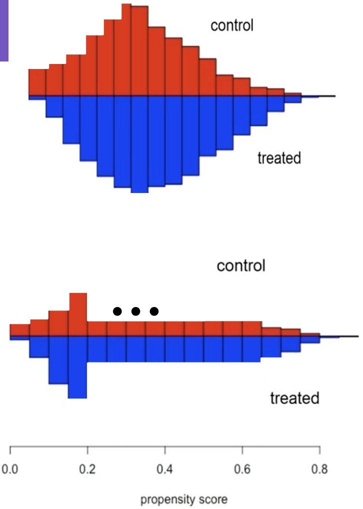
Propensity Score Matching on Zip Code Level



Propensity Score Model



Group users by zip codes



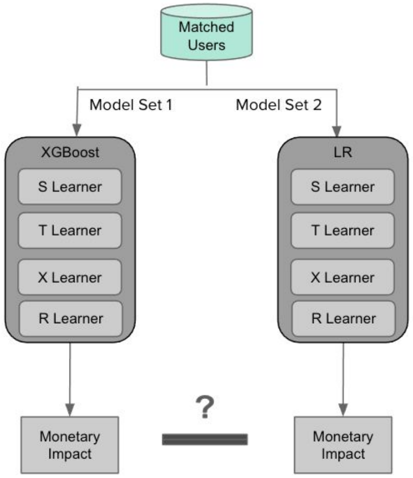
Propensity Score Matching by zip codes

```
rom causalml.match import NearestNeighborMatch

atcher = NearestNeighborMatch(caliper=0.1, replace=True)
f_matched = matcher.match_by_group(data=df,
                                   treatment_col=TREATMENT_COL,
                                   score_cols=[SCORE_COL],
                                   groupby_col='zip_code')
```

CeViChE Framework

Inference: Leveraging different methods to validate inference

- Apply models on the matched dataset to estimate Average Treatment Effect (ATE). This process is cross-validated by various algorithms and time frames.
- Meta-learner algorithms[1], [2] as the main inference model to estimate the impact of the conversion event on gross bookings.
    - S Learner
    - T Learner
    - X Learner
    - R Learner
- Evaluate the inference validity by two sets of models using XGBoost and Linear Regression as base learner.



[1] Kunzel S, Sekhon J, Bickel P, Y Bin. “Meta-learners for Estimating Heterogeneous Treatment Effects using Machine Learning”, arXiv preprint arXiv:1706.03461.
[2] Alaa A, van der Schaar M. “Limits of Estimating Heterogeneous Treatment Effects: Guidelines for Practical Algorithm Design”, AAAI 2018.

CeViChE Framework

Sensitivity Analysis to Check the Robustness

Placebo Treatment

Replace the treatment with a random variable, then retrain the meta-learners

Replace/Add Irrelevant Confounder

Add/replace a random variable to introduce noise to the system, then retrain the meta-learners

Subset Validation

Remove a random subset of the data, then retrain the meta-learners

Selection Bias[1]

One Sided confounding function and Alignment confounding function

```python
from sklearn.linear_model import LinearRegression
from causalml.inference.meta import BaseXLearner
from causalml.metrics.sensitivity import Sensitivity

learner = BaseXLearner(LinearRegression())
sens = Sensitivity(df=df, inference_features=INFERENCE_FEATURES, p_col=SCORE_COL,
                   treatment_col=TREATMENT_COL, outcome_col=OUTCOME_COL, learner=learner)

# check the sensitivity summary report
sens_summary = sens.sensitivity_analysis(methods=['Placebo Treatment',
                                                  'Random Cause',
                                                  'Subset Data',
                                                  'Random Replace',
                                                  'Selection Bias'], sample_size=0.5)
```

Method	ATE	New ATE	New ATE LB	New ATE UB
Placebo Treatment	0.4552	0.0174	-0.0387	0.0736
Random Cause	0.4552	0.4546	0.3994	0.5098
Subset Data(sample size @0.9)	0.4552	0.4477	0.3901	0.5054
Random Replace	0.4552	0.4549	0.3997	0.5101
Selection Bias (alpha@-0.6281, with r-sqaure:0...	0.4552	1.0816	1.0264	1.1368
Selection Bias (alpha@-0.50248, with r-sqaure:...	0.4552	0.9563	0.9011	1.0115
Selection Bias (alpha@-0.37686, with r-sqaure:...	0.4552	0.8310	0.7758	0.8862
Selection Bias (alpha@-0.25124, with r-sqaure:...	0.4552	0.7058	0.6506	0.7610
Selection Bias (alpha@-0.12562, with r-sqaure:...	0.4552	0.5805	0.5253	0.6357
Selection Bias (alpha@0.0, with r-sqaure:0.0	0.4552	0.4552	0.4000	0.5104
Selection Bias (alpha@0.12562, with r-sqaure:0...	0.4552	0.3300	0.2748	0.3852
Selection Bias (alpha@0.25124, with r-sqaure:0...	0.4552	0.2047	0.1495	0.2599
Selection Bias (alpha@0.37686, with r-sqaure:0...	0.4552	0.0794	0.0242	0.1346
Selection Bias (alpha@0.50248, with r-sqaure:0...	0.4552	-0.0458	-0.1010	0.0094
Selection Bias (alpha@0.6281, with r-sqaure:0....	0.4552	-0.1711	-0.2263	-0.1159

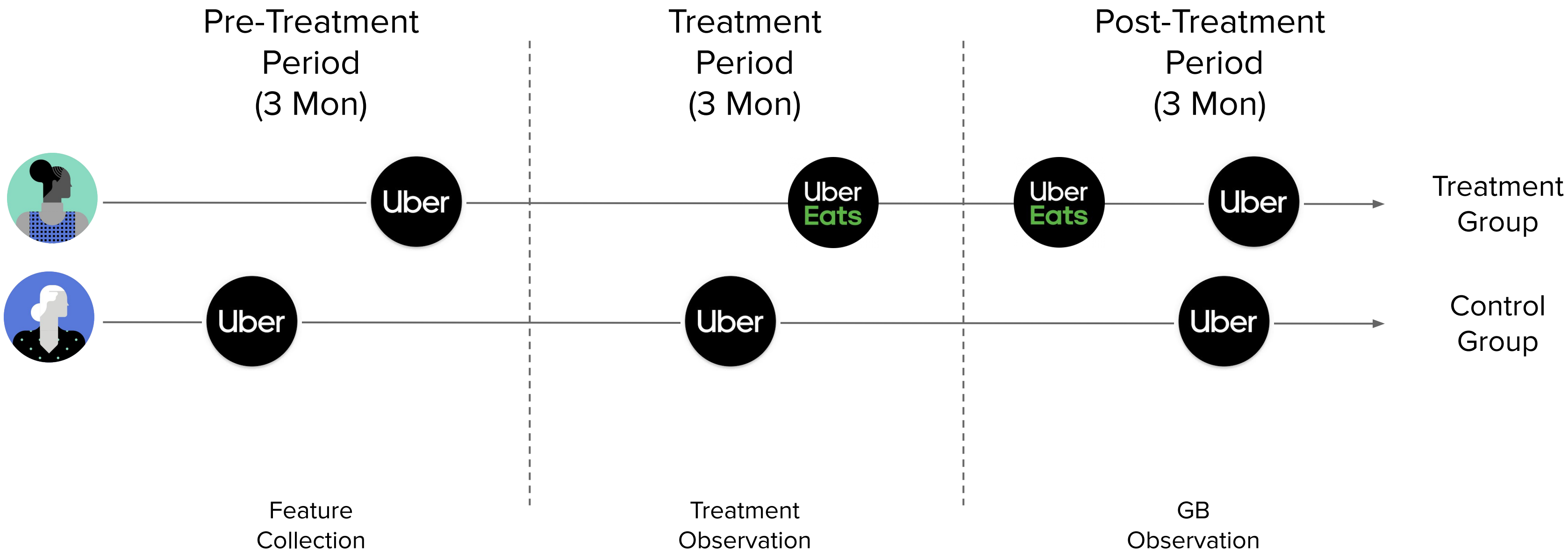
[1] Blackwell, Matthew. "A selection bias approach to sensitivity analysis for causal effects." Political Analysis 22.2 (2014): 169-182.

Case Study

Monetary Impact of to Eater Rider (R2E) Cross Sell

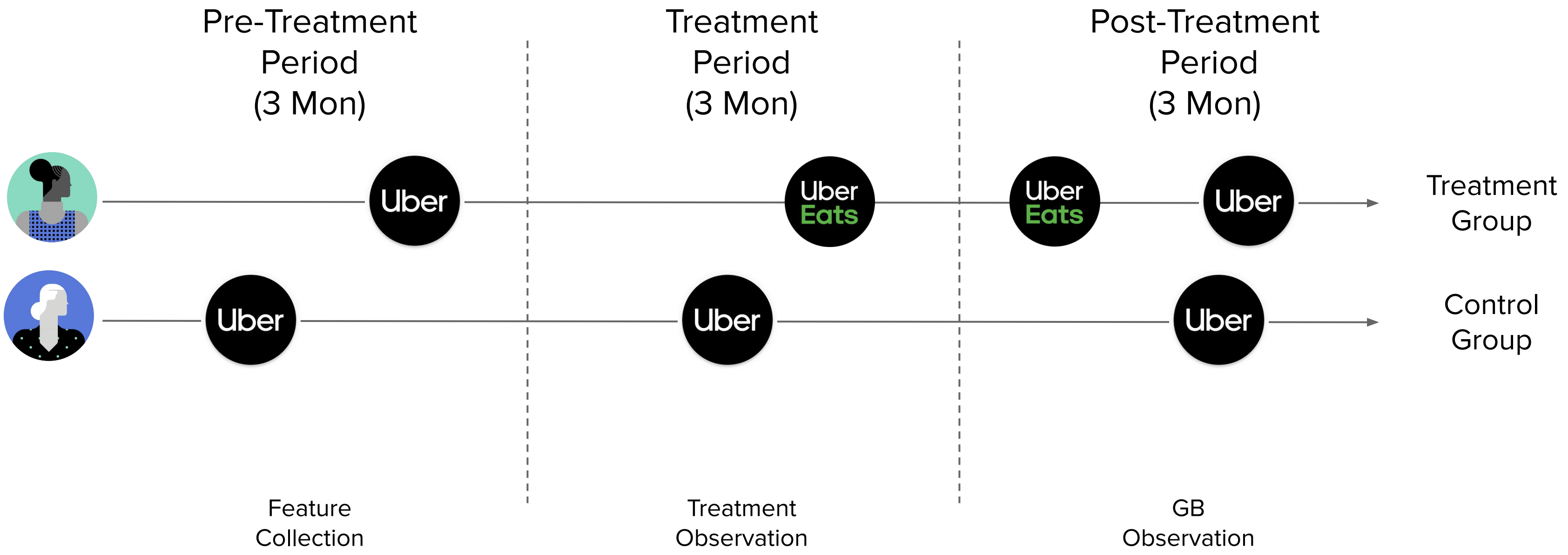
Notebook: t.uber.com/kdd_causalml_case1

Analysis Setup



- Treatment Group: Riders who converted to Eaters within treatment period
- Control Group: Riders who were not converted to Eaters until the end of post-treatment period

Analysis Setup



- Treatment Group: Riders who converted to Eaters within treatment period
- Control Group: Riders who were not converted to Eaters until the end of post-treatment period

Propensity Score Matching

Matching removes biases between treatment and control

Before matching

Treatment: **4K users**
R2E Conversion

- 3 Mon GB*: **$2,000**
- 3 Mon Trips: 100

Control: **96K users**
No R2E

- 3 Mon GB: **$20**
- 3 Mon Trips: 10

After matching

Treatment: **4K users**
R2E Conversion

- 3 Mon GB: **$2,000**
- 3 Mon Trips: 100

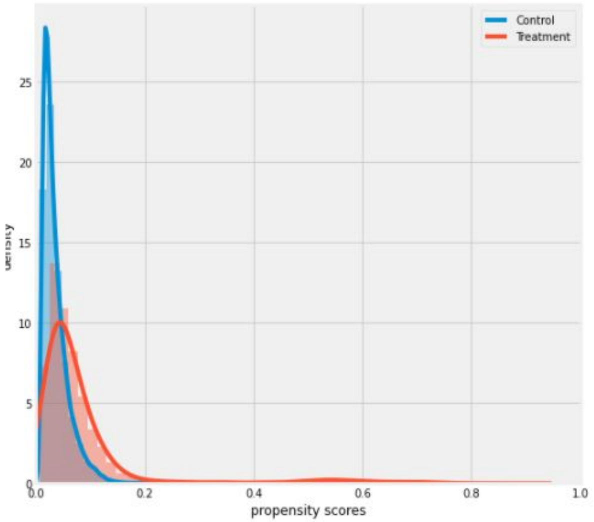
Control: **4K users**
No R2E

- 3 Mon GB: **$2,000**
- 3 Mon Trips: 100

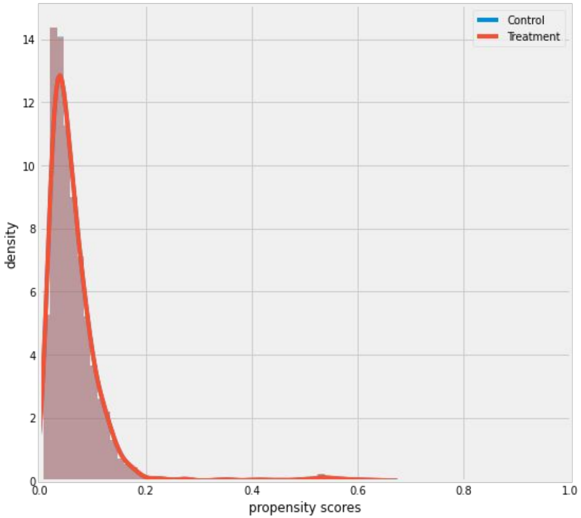
* Number here just for illustration purpose not from the real Uber Rider datasets

Matching Validation

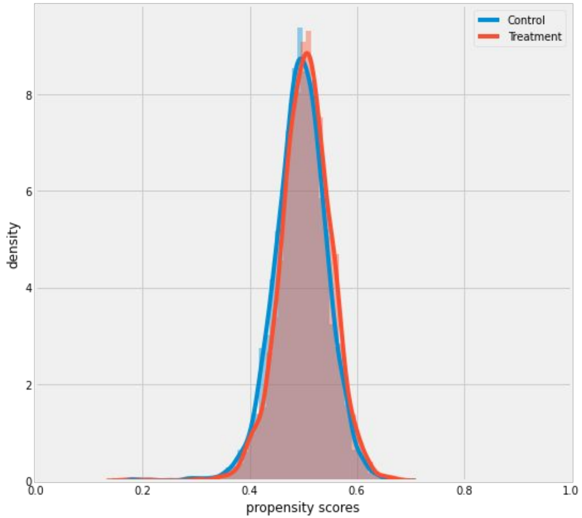
Density Plot of Propensity Score before Matching



Density Plot of Propensity Score after Matching



Density Plot of Propensity Score after Recalibrating



Matching Validation

Standard Mean Difference (SMD) on observed covariates, with |SMD| < 0.1 considered balanced

100+ covariates:
Static,
LTV,
Trips/Orders,
Marketing,
Comms,
Promo
...

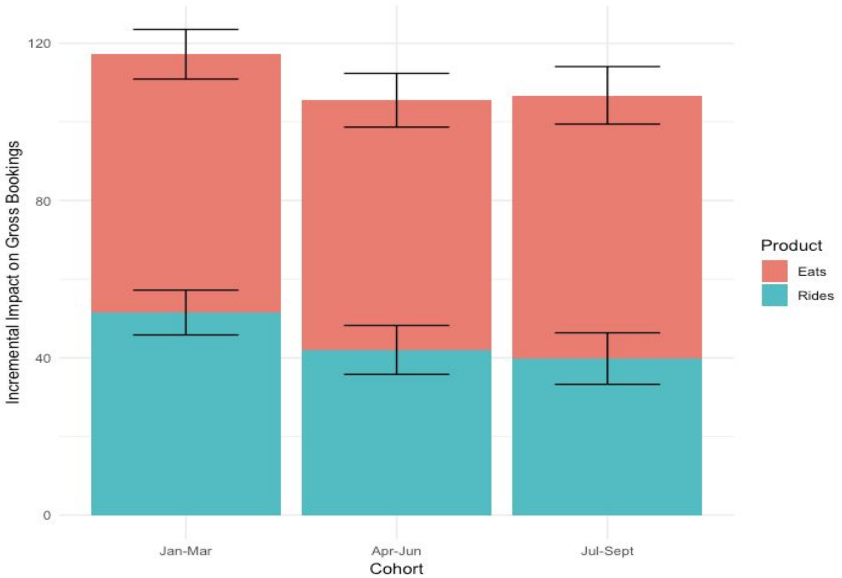
	Pre-Matching			After-Matching		
Variable	**Control**	**Treatment**	**SMD**	**Control**	**Treatment**	**SMD**
n	96k	4k		96k	4k	
feature_1	0.11 (0.17)	0.30 (0.13)	**1.248**	0.30 (0.13)	0.30 (0.13)	0.000
feature_2	62.99 (137.07)	17.28 (51.43)	**-0.442**	16.94 (51.15)	17.28 (51.43)	0.007
feature_3	0.53 (1.84)	2.14 (3.57)	**0.568**	2.14 (3.57)	2.14 (3.57)	-0.001
feature_4	0.87 (1.49)	1.97 (2.98)	**0.470**	1.99 (3.02)	1.97 (2.98)	-0.005
feature_5	0.80 (3.18)	3.61 (7.87)	**0.467**	3.53 (7.73)	3.61 (7.87)	0.009
feature_6	0.08 (0.12)	0.31 (0.17)	**1.538**	0.31 (0.17)	0.31 (0.17)	0.000
fefeature_7	1.76 (2.21)	4.60 (1.71)	**1.435**	4.62 (1.69)	4.60 (1.71)	-0.011
...	...	...	...	...	...	...

Meta Learners

Consistent Results across Meta Learners

	S Learner (LR)	S Learner (XGB)	T Learner (LR)	T Learner (XGB)	X Learner (LR)	X Learner (XGB)	R Learner (LR)	R Learner (XGB)
ATE	0.4473	0.3964	0.4479	0.4405	0.4479	0.4437	0.4458	0.4450
Lower	0.3923	0.3478	0.3928	0.3933	0.3928	0.3975	0.4451	0.4444
Upper	0.5024	0.4450	0.5030	0.4876	0.5030	0.4899	0.4465	0.4457
Baseline	0.2896	0.3405	0.2891	0.2964	0.2891	0.2932	0.2911	0.2919
Lift	1.5446	1.1641	1.5495					

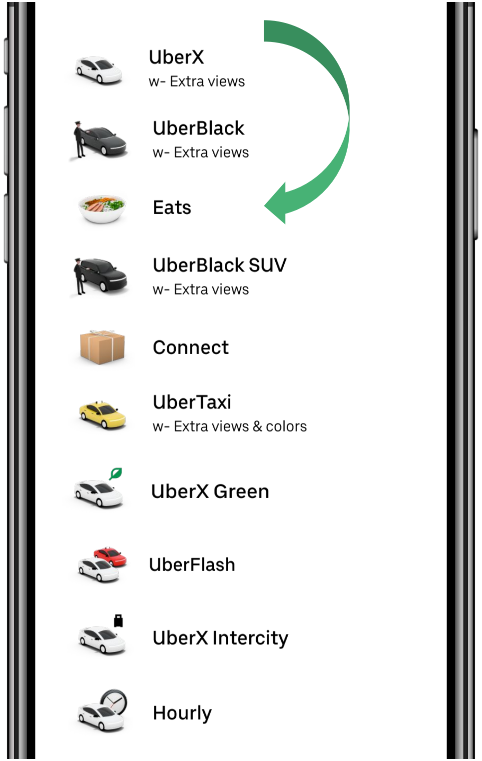
Cross-sell drives synergistic lift:
Eats conversion drives Rides spend



Sensitivity Analysis

Robust Results Validated by Sensitivity Analysis

Method	ATE	New ATE	New ATE LB	New ATE UB
Placebo Treatment	0.4552	0.0174	-0.0387	0.0736
Random Cause	0.4552	0.4546	0.3994	0.5098
Subset Data(sample size @0.9)	0.4552	0.4477	0.3901	0.5054
Random Replace	0.4552	0.4549	0.3997	0.5101
Selection Bias (alpha@-0.6281, with r-sqaure:0...	0.4552	1.0816	1.0264	1.1368
Selection Bias (alpha@-0.50248, with r-sqaure:...	0.4552	0.9563	0.9011	1.0115
Selection Bias (alpha@-0.37686, with r-sqaure:...	0.4552	0.8310	0.7758	0.8862
Selection Bias (alpha@-0.25124, with r-sqaure:...	0.4552	0.7058	0.6506	0.7610
Selection Bias (alpha@-0.12562, with r-sqaure:...	0.4552	0.5805	0.5253	0.6357
Selection Bias (alpha@0.0, with r-sqaure:0.0	0.4552	0.4552	0.4000	0.5104
Selection Bias (alpha@0.12562, with r-sqaure:0...	0.4552	0.3300	0.2748	0.3852
Selection Bias (alpha@0.25124, with r-sqaure:0...	0.4552	0.2047	0.1495	0.2599
Selection Bias (alpha@0.37686, with r-sqaure:0...	0.4552	0.0794	0.0242	0.1346
Selection Bias (alpha@0.50248, with r-sqaure:0...	0.4552	-0.0458	-0.1010	0.0094
Selection Bias (alpha@0.6281, with r-sqaure:0....	0.4552	-0.1711	-0.2263	-0.1159



Summary & Takeaways

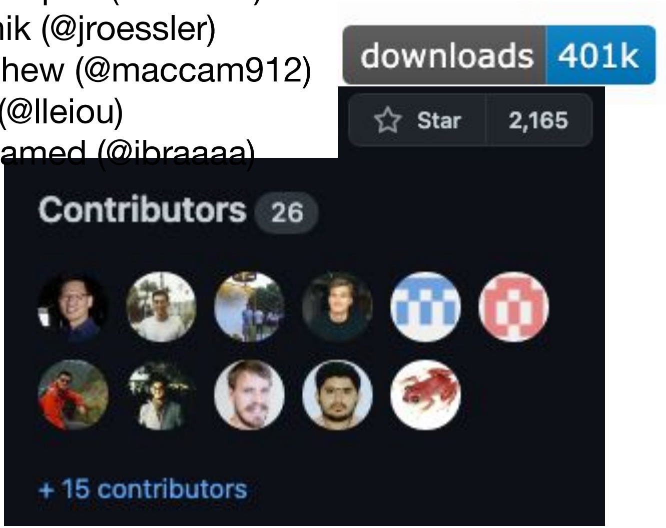
Summary | Core team and Contributors

Shoutout to all the contributors[1] of CausalML community!

Dahee Lee
Huigang Chen
Jeong-Yoon Lee
Jing Pan
Mike Yung
Mert Bay
Paul Lo
Totte Harinen
Yifeng Wu
Zhenyu Zhao

Yuchen (@yluogit)
Manoj (@manojbalaji1)
Peter (@peterfoley)
Suraj (@surajiyer)
Harsh (@HarshCasper)
Fritz (@fritzo)
Tomasz (@TomaszZamacinski)
Georg (@waltherg)
Florian (@FlorianWilhelm)
Harry (@deeplaunch)
Katherine (@khof312)

Steve (@steveyang90)
Vaclavbelak (@vaclavbelak)
Mario (@mwijaya3)
Christophe (@ccrndn)
Jannik (@jroessler)
Matthew (@maccam912)
Leo (@lleiou)
Mohamed (@ibraaaa)

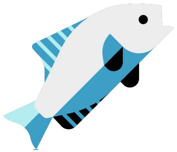


[1] List of contributors: https://github.com/uber/causalml/graphs/contributors

Q&A

Key Takeaways

- **Recommendations:** promote more Rider-to-Eater campaigns to drive the cross-sell conversions as it have incremental monetary impact for both products
- Applied in **multiple projects** at Uber to measure and predict the incrementality of cross-sell, disengagement, loyalty program, and marketing campaigns
- **Observational data** can be used to establish the causal relationship when AB test fails; proposed CeViChE framework creates a **scalable and flexible framework** that establishes causality between user actions and monetary value



- **CausalML** provides one-stop shop for an end-to-end solution
    - Notebook: t.uber.com/kdd_causalml_case1

Appendix