

BREAK  
THROUGH  
TECH

# Unwrapping Customer Delight

## Milestone #5 Meeting: The Analysis Phase

The Estée Lauder Companies

October 31, 2025



# From Unadjusted ATE → MLRATE

- So far (Unadjusted ATE):

$$y = \beta_0 + \beta_1 T + \varepsilon \Rightarrow \hat{\beta}_1 = \text{ATE}$$

Randomization  $\Rightarrow$  unbiased difference in means.

- **Why go beyond an unadjusted ATE?**

- Outcomes vary for many reasons unrelated to treatment (baseline behavior, seasonality, customer heterogeneity).
- If we can **predict baseline outcomes** using **pre-treatment covariates**, we can **explain away variance** and get **narrower Confidence Intervals** (more power) *without* more data.

- **MLRATE idea:** use ML to predict baseline outcome  $G(x)$  from **pre-treatment features**, then **adjust** the ATE regression with  $G(x)$  to reduce residual noise.





# MLRATE Intuition

- **Step 1 (Predict baseline):** Train an ML model on *all* customer's pre-treatment features  $x$  to get  $G(x_i) \approx E[Y | x_i, \text{no treatment}]$  - the predicted outcome under business-as-usual conditions.  
→ Use **out-of-fold** predictions to avoid data leakage: keeps predictions ind. of noise and treatment effect estimates unbiased

- **Step 2 (Variance reduction in ATE regression):**

$$y_i = \beta_0 + \beta_1 T_i + \beta_2 G(x_i) + \beta_3 T_i (G(x_i) - \bar{G}) + \varepsilon_i$$

- $G(x_i)$  “soaks up” explainable outcome variation (reduces residual variance)
- Centering  $G(x_i)$  makes  $\beta_1$  interpretable as the ATE at the average baseline and reduces collinearity with the intercept
- Randomization ensures  $T$  is independent of  $x$  on average  $\Rightarrow$  adjustment improves precision without inducing bias
- **Why this helps:** If  $G(x)$  is predictive (high  $R^2$ ), the residuals shrink  $\Rightarrow$  smaller SEs  $\Rightarrow$  tighter CIs  $\Rightarrow$  (typically) higher power
- Mental model: Same causal target as Unadjusted ATE; MLRATE just denoises the outcome first using valid pre-treatment information





# Implementing MLRATE

```
import statsmodels.api as sm

# 1) Obtain out-of-fold predictions of baseline outcome
#     Reuse your K-fold routine to get leakage-free predictions:
g_hat = out_of_fold_predict_baseline(X, y, model="RandomForest")

# 2) Center the predictions for interpretation & stability
Gbar = g_hat.mean() # "G_mean" in the slides
G_centered = g_hat - Gbar # defines "average baseline" as zero so that
                          # T coefficient is the ATE at average baseline

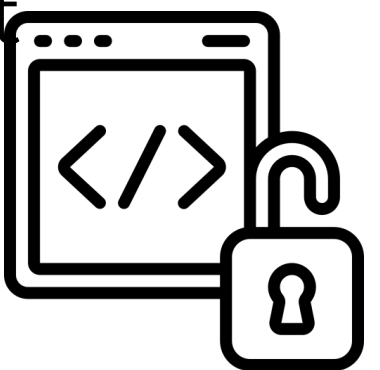
# 3) Build the regression design for ATE with variance reduction
X_reg = add_constant(np.column_stack([T, g_hat, T * G_centered]))

# columns: [const, T, G, T*(G-Gbar)]

# 4) Fit with robust SEs and extract ATE & CI
ols = OLS(y, X_reg).fit(cov_type="HCO")
ate = model.params[1]
ci_lower, ci_upper = model.conf_int()[1]

print(f"MLRATE ATE: {ate:.4f}")
print(f"95% CI: [{ci_lower:.4f}, {ci_upper:.4f}]" )
```

- To keep in mind:
  - Any reasonable ML regressor works (RF/GBM/linear), as long as it uses **only pre-treatment features** (or any features uncorrelated with treatment assignment) and predictions are **out-of-fold**
  - Use robust (HC) SEs
  - If  $G(x)$  is weak (low predictive power - measured by out-of-sample  $R^2$ ), expect little gain vs. OLS; if strong, expect **meaningful CI shrinkage**

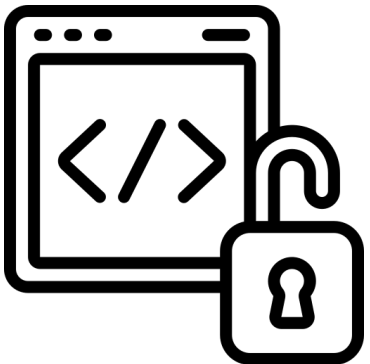






# Implementing MLRATE

```
--- MLRATE Adjusted ATE ---
                                OLS Regression Results
=====
Dep. Variable:          revenue (t)    R-squared:
Model:                  OLS           Adj. R-squared:
Method:                 Least Squares  F-statistic:
Date:                   Thu, 30 Oct 2025  Prob (F-statistic):
Time:                   17:28:29        Log-Likelihood:
No. Observations:      5556           AIC:
Df Residuals:          5552           BIC:
Df Model:               3
Covariance Type:       HC0
=====
                                coef    std err          z      P>|z|      [0.025      0.975]
-----
const
T
g_hat
T * (g_hat - g_bar)
=====
Omnibus:
Prob(Omnibus):
Skew:
Kurtosis:
=====
```



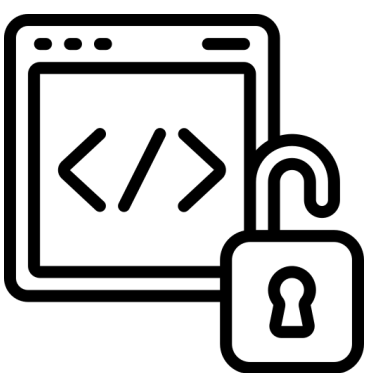


# Comparing MLRATE vs Unadjusted ATE

Aspect	Unadjusted ATE	MLRATE
Target	Same ATE	Same ATE
Inputs	$T$ only	$T, G(x), T \times (G(x) - G_{\text{mean}})$
Bias (RCT)	Unbiased	Unbiased
Variance / CI	Larger/wider	Smaller/narrower (if $G$ predictive)
When it shines	Baseline	Noisy outcome, cost constraints


- **Questions to guide your analysis:**

- How much did your **CI width** change from OLS to MLRATE?
- Is  $G(x)$  actually predictive (what's its  $R^2$  vs.  $y$ )?
- Does the **p-value** for the treatment coefficient change meaningfully? Why?
- Business lens: does higher precision change your decision (launch gift strategy vs. hold)?
- Any signs of leakage or overfitting (did you truly use out-of-fold predictions)?





# Estimating MLRATE Adjusted ATE

- **Generate baseline predictions**
  - Use your pre-treatment features to build an ML model that predicts baseline (business-as-usual outcomes)
  - Obtain out-of-fold predictions  $G(x)$  to avoid data leakage
- **Estimate the MLRATE ATE**
  - Center predictions (i.e define  $G_{\text{mean}}$ )
  - Fit model according to slides 3 and 4
  - Extract  $\beta_1$   MLRATE ATE
    - Also 95% confidence interval and p-value
- **Compare to your Unadjusted ATE** (see slide 6 for more guiding questions)
  - How do the two points estimates differ? Did your CI narrow? etc.
- **Reflect** (see slide 5):
  - Is your baseline model  $G(x)$  actually predictive? When might MLRATE offer little or no improvement? etc.
  - Interpret all fitted coefficients and their p-values in the new MLRATE OLS model.
- **Meeting will also cover:**
  - Setting you up for the final project milestone :)





# Project milestones and timeline

These are the milestones for your Challenge Project. They include the [CRISP-DM](#) process steps you learned about in your ML Foundations course. In addition, there is an educational component in the front-end.

