Estée Lauder 1B

Unwrapping Customer Delight: Optimizing Surprise Gift Strategies with Randomized Controlled Trials and MLRATE

DECEMBER 2025

AI Studio Final Project

BREAK THROUGH TECH

Introduction

Meet Our Team

Sehr Abrar
The City College of New York

Lydia Aubourg
Rutgers University

Ava Leung
Rutgers University

Meet Our Team

Charitha Sarraju
The University of Texas at Dallas

Sandy Wu
Stony Brook University

Xiaoyan Li
Hunter College

Nidhi Parvathala
The University of Texas at Dallas

Our AI Studio Coach & Challenge Advisors

Ananya Devarakonda
AI Studio Coach

Eddy El Chaar
Challenge Advisor

Luis Saldaña
Challenge Advisor

Presentation Agenda

Introduction	2 minutes
AI Studio Project Overview	5 minutes
Literature & Pre-Experiment Data Exploration	5 minutes
Power Analysis & Experiment Data Exploration	5 minutes
Regression Modeling & Treatment Effect	5 minutes
Final Conclusions	2 minutes
Questions	20 minutes

AI Studio Project Overview

Estée Lauder

The **Estée Lauder Companies (ELC)** is a **global leader in prestige beauty**, manufacturing, marketing, and selling high-quality skincare, makeup, fragrance, and hair care products for luxury and prestige consumers worldwide.

- **Values:** Generosity of spirit, fearless persistence, highest aesthetic standards, respect for the individual, uncompromising quality, and ethics & integrity.
- **Brands:** Owns 20+ beauty brands including MAC, Clinique, La Mer, Bobbi Brown, and Estée Lauder, reaching consumers in ~150 countries and territories.
- **Fun Fact:** 81% of ELC's global workforce are women, and the company remains headquartered in New York City with 60,000+ employees worldwide.

ESTĒE LAUDER

To measure the impact of surprise gifts on customer spending using machine learning and provide insights for marketing ROI.

Our Goal

1. **Design** and run a Randomized Controlled Trial (RCT) to test the effect of surprise gifts.
2. **Apply** the MLRATE method to precisely measure the causal impact of gifts.
3. **Build** machine learning models to predict customer spending from pre-experiment data.
4. **Analyze** actual experimental data to evaluate spending behavior.
5. **Quantify** the ROI of surprise gift campaigns for business insights.
6. **Provide** recommendations to optimize future gift-giving strategies.

Business Impact

- **ROI:** Deliver a robust quantification of the financial impact of surprise gift campaigns.
- **Optimization:** Generate insights to improve and optimize future gift-giving strategies.
- **Validation:** Compare estimated treatment effects to the simulated true effect for accuracy.
- **Strategic Insights:** Support data-driven decisions to enhance marketing effectiveness and customer engagement.

Our Approach

SEPT

Review key resources on regression, RCTs, and MLRATE.

Perform exploratory data analysis (EDA) on pre-experiment data and present findings.

OCT

Conduct power analysis to determine required sample sizes using standard and MLRATE methods and compare the results.

Perform EDA on experimental data to check data quality, explore distributions, compare treatment vs. control groups, and visualize potential treatment effects.

NOV

Fit an OLS model to estimate the treatment effect and interpret the ATE, confidence intervals, and p-values.

Use ML to generate baseline predictions and apply MLRATE to estimate the adjusted treatment effect, compare to the unadjusted ATE, and interpret results.

DEC

Finalize results by summarizing RCT and MLRATE analyses.

Quantify the impact of surprise gifts and provide recommendations for future campaigns.

Resources We Used

pandas

NumPy

matplotlib

statsmodels

seaborn

SciPy

scikit learn

Literature & Pre-Experiment Data Exploration

Key Vocabulary

- **Randomized Controlled Trial (RCT)** → An experiment where participants are randomly assigned to treatment or control groups to measure causal effects.
- **MLRATE** → Machine Learning Regression-Adjusted Treatment Effect Estimator; improves precision of treatment effect estimates using ML predictions in OLS regression.
- **Average Treatment Effect (ATE)** → The expected difference in outcomes between treatment and control groups; quantifies the causal impact of an intervention.
- **Variance Reduction** → Methods that reduce variability in estimates to improve statistical power and accuracy of treatment effect measurement.
- **Power Analysis** → Statistical calculation used to determine the sample size needed to reliably detect an effect.
- **Covariates** → Pre-intervention variables used to predict outcomes and adjust treatment effect estimates (e.g., prior spending, demographics).
- **Cross-Fitting** → ML technique where predictions are generated on separate data folds to prevent overfitting and bias in treatment effect estimation.

Pre-Experiment Data

Data Provided (pre-experiment_data.parquet):

- **Numerical**: Customer ID, Total Prior Spending, Avg Transaction Value, Purchase Frequency, Recency of Last Purchase
- **Categorical**: Customer Tenure, Loyalty Program Status

EDA Purpose:

- Explore distributions, patterns, and potential relationships before RCT
- Check for missing values, outliers, and inconsistencies

Outcome:

- Generated summary statistics and visualization
- Cleaned and preprocessed data for downstream analysis

Pre-Experiment Data

Main Findings

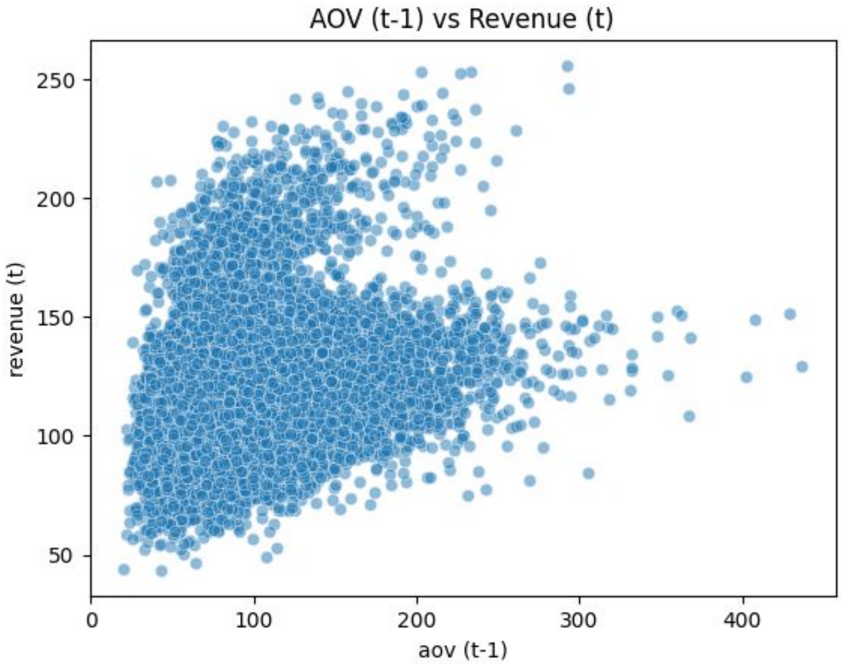
- Majority of customers are low or moderate spenders; a small elite group drives most revenue (mean = $117).
- Past AOV is a strong predictor of current revenue, higher past spenders remain high-value.
- Loyalty members generate significantly higher revenue (median ≈ $175 vs $110 for non-members).
- Tenure_in_days has negligible effect compared to spending behavior.

Conclusion:

- Revenue is driven by past spending and loyalty engagement

Pre-Experiment Data

EDA - AOV vs Revenue



Pre-Experiment Data

EDA - Revenue by Loyalty Membership

Power Analysis & Experiment Data Exploration

Power Analysis

What is a Power Analysis?

- **Definition**: Planning tool to determine the sample size needed to detect a minimum detectable effect (MDE) with a chosen power and significance level.
- **Why It Matters**: too few users → miss real effects; too many → waste resources.
- **Key Inputs**: Power ($1-\beta$), Significance (α), MDE, and Variance.
- **Project Baseline**: we use 90% power, α = 5%, MDE = 1% of mean revenue

Power Analysis

Standard vs. MLRATE Power Analysis

Standard Power Analysis

- Uses a two-sample t-test to detect an average treatment effect.
- **Requires:** MDE, significance (α), power (1−β), and variance of outcome (Y).
- **Effect size = MDE / sd(Y) → Solve for required sample size (N).**

MLRATE (Machine Learning Regression Average Treatment Effect)

- Extends standard power analysis by first removing predictable variation in outcomes.

1. Use ML (e.g., Random Forest or GBM) to predict $\hat{Y} = G(X)$ from pre-treatment features.
2. Compute residuals: $\varepsilon = Y - \hat{Y}$.
3. Fit OLS: $Y \sim 1 + G(X)$ to estimate the effect.
4. Use sd(ε) (residual variance) in the power calculation instead of sd(Y).

- Result: Lower variance → higher precision → smaller sample size needed for same MDE.

Power Analysis

Method 1: Standard Power Analysis (T-Test)

- Required sample size per group: 8,801 (≈17,600 total)
- Large sample size because small effects are harder to detect due to natural variance in revenue.

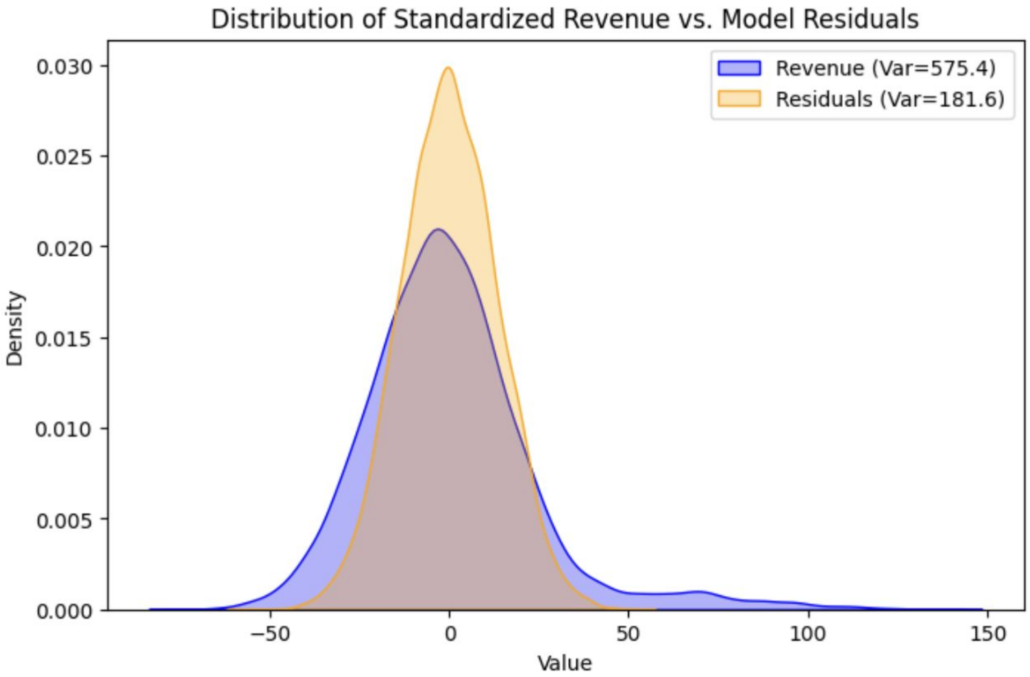
Method 2: MLRATE Power Analysis (Variance-Based)

- Required sample size per group: 2,778 (≈5,556 total, smaller amount than T-Test result)
- Smaller sample size because adjusting for covariates reduces unexplained variance, making small effects easier to detect.

MLRATE reduces the required sample size, demonstrating the benefit of variance reduction through covariate adjustment.

Power Analysis

Recreation of the Residual Distribution Plot



Experiment Data

Main Findings

Distribution Patterns:

- Right-skewed revenue with high-value customer tail
- Most customers are recent purchasers; small segment with purchase gaps

Key Relationships:

- Revenue positively correlates with prior AOV
- Negative relationship with recency (recent buyers spend more)

Treatment Effects:

- Subtle but statistically significant overall effects
- Stronger impact observed for loyalty members and high-AOV segments

Data Quality:

- Outliers primarily among high-value customers (expected)

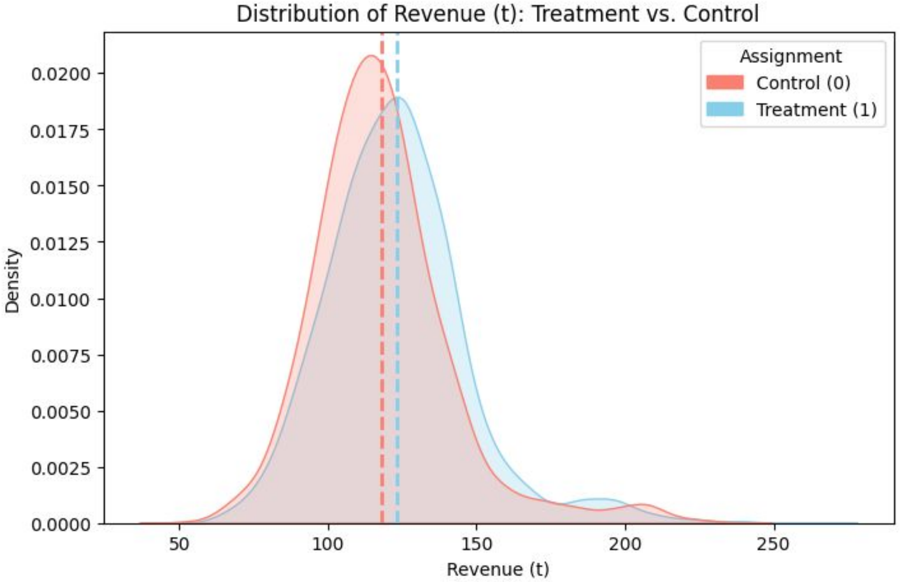
Experiment Data

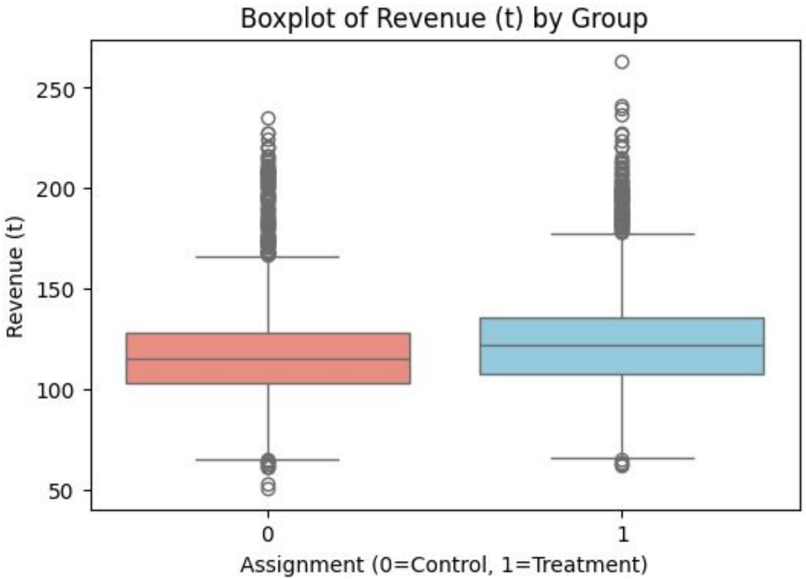
EDA - Compare distributions of revenue (t) between groups

	Covariate	Mean_Treatment	Mean_Control	SMD	p-value
0	aov (t-1)	97.09	97.86	-0.019	0.4747
1	days_since_last_purchase (t-1)	14.34	14.75	-0.027	0.3145
2	tenure_in_days(t-1)	49.40	49.81	-0.006	0.8119
3	loyalty_membership	0.07	0.06	0.014	0.6060

Experiment Data

EDA - Compare distributions of revenue (t) between treatment and control





Experiment Data

EDA - Compare distributions of revenue (t) between treatment and control

assignment	mean	median	std	count
Control	117.952016	115.5700	23.870081	2773
Treatment	123.053486	121.8105	24.174141	2783

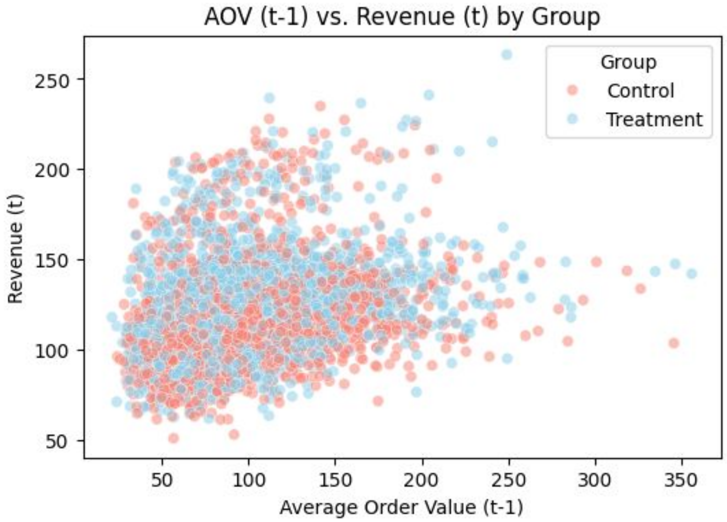
Experiment Data

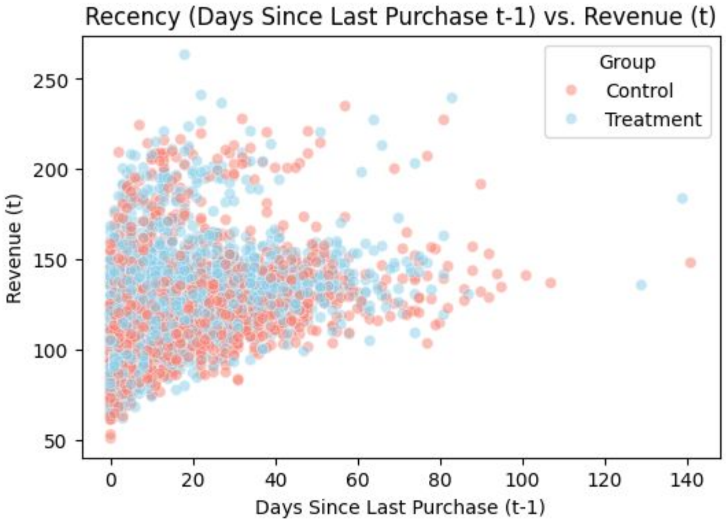
EDA - Perform an independent samples t-test

```
Independent Samples t-test Results:
  t-statistic: 7.9146
  p-value: 0.0000

Conclusion: Reject the null hypothesis. There is a statistically significant difference in mean revenue between the treatment and control groups.
```

Experiment Data

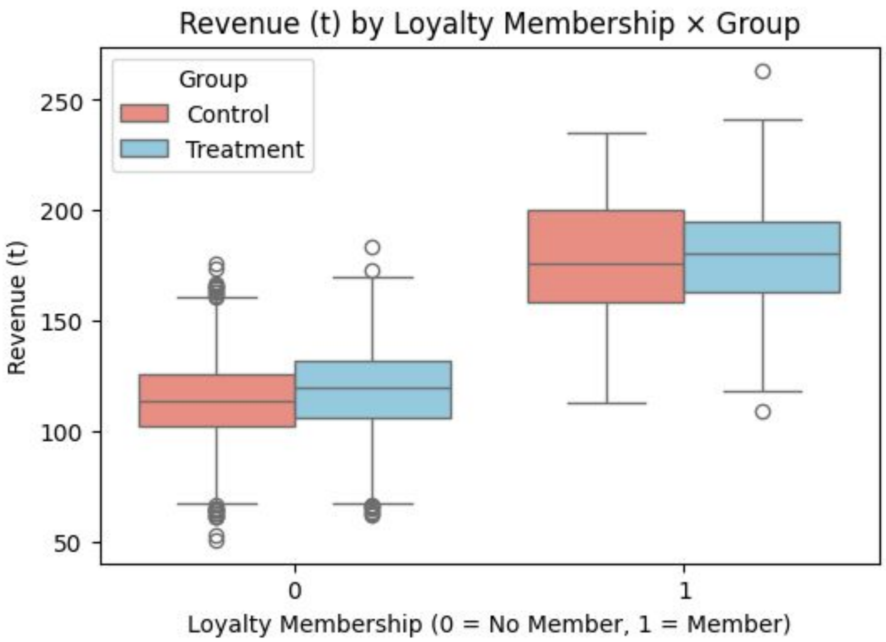
EDA - Explore relationships between key variables and revenue (t)





Experiment Data

EDA - Explore relationships between key variables and revenue (t)



Experiment Data

EDA - Analyze & Understand Skews & Outliers

```
Skewness of numeric features:
aov (t-1): 1.26
days_since_last_purchase (t-1): 2.00
tenure_in_days(t-1): 5.06
revenue (t): 1.07

Outlier Summary (IQR method):
                        Feature  Num Outliers  % Outliers  Lower Bound  \
0                     aov (t-1)           165    2.969762     -3.93625
1  days_since_last_purchase (t-1)           277    4.985601    -20.00000
2            tenure_in_days(t-1)           428    7.703384    -51.00000
3                   revenue (t)           213    3.833693     64.53825

   Upper Bound
0    190.53375
1     44.00000
2    125.00000
3    172.89825
```

Regression Modeling & Treatment Effect

Estimating the ATE

Goal: Quantify the causal effect of the "gift" campaign on post-intervention revenue.

Approach: Use OLS regression to estimate the unadjusted Average Treatment Effect (ATE).

Model:

$$\mathrm{Revenue} = \beta_0 + \beta_1 \times \mathrm{Treatment} + \varepsilon$$

where β_1 = ATE.

Key Insight: Since treatment was randomized, differences in average revenue between treated and control groups reflect the campaign's causal impact.

Estimating the ATE

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            revenue (t)   R-squared:                       0.011
Model:                            OLS   Adj. R-squared:                  0.011
Method:                 Least Squares   F-statistic:                     62.66
Date:                Fri, 31 Oct 2025   Prob (F-statistic):           2.94e-15
Time:                        13:43:29   Log-Likelihood:                -25545.
No. Observations:                5556   AIC:                         5.109e+04
Df Residuals:                    5554   BIC:                         5.111e+04
Df Model:                           1
Covariance Type:                  HC0
=========================================================================================
                            coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                   117.9520      0.453    260.258      0.000     117.064     118.840
treatment (gift sent)     5.1015      0.644      7.916      0.000       3.838       6.365
==============================================================================
Omnibus:                     1111.304   Durbin-Watson:                   2.009
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2882.493
Skew:                           1.083   Prob(JB):                         0.00
Kurtosis:                       5.785   Cond. No.                         2.62
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)

estimated ate: $5.10
95% ci: [3.84, 6.36] (width = 2.53)
p-value: 0.0000
```

- Campaign increased spending by ~$5.10, showing a clear positive impact.
- 95% CI [3.84, 6.36] is narrow and excludes 0 → result is precise and reliable.
- p-value ≈ 0 → strong evidence against the null; effect is significant.
- Large sample (5,556) improved precision; smaller samples or higher variance would widen the CI.

Why MLRATE?

The Challenge:

- Standard ATE: $5.10 with CI [3.84, 6.36]
- $R^2 = 1.1\%$ — minimal variance explained
- High unexplained variance limits precision

The MLRATE Approach:

- Leverages ML to predict revenue from customer covariates
- Removes predictable variance before estimating treatment effect
- Increases R^2 from 1.1% to 67.1%

The Result:

- MLRATE ATE: $5.32 with CI [4.59, 6.05] (42% narrower)
- Standard error reduced by 42%
- Future sample size requirements reduced by 68%

Implementing MLRATE

```
Model Summary:
                            OLS Regression Results
==============================================================================
Dep. Variable:            revenue (t)   R-squared:                       0.671
Model:                            OLS   Adj. R-squared:                  0.671
Method:                 Least Squares   F-statistic:                     3644.
Date:                Thu, 13 Nov 2025   Prob (F-statistic):               0.00
Time:                        19:18:37   Log-Likelihood:                -22484.
No. Observations:                5556   AIC:                         4.498e+04
Df Residuals:                    5552   BIC:                         4.500e+04
Df Model:                           3
Covariance Type:                  HC0
=========================================================================================
                          coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                   3.6446      1.562      2.333      0.020       0.583       6.706
T                       5.3194      0.371     14.322      0.000       4.591       6.047
g_hat                   0.9470      0.013     73.987      0.000       0.922       0.972
T * (g_hat - g_bar)    -0.0030      0.018     -0.163      0.871      -0.039       0.033
==============================================================================
Omnibus:                        5.447   Durbin-Watson:                   1.986
Prob(Omnibus):                  0.066   Jarque-Bera (JB):                6.046
Skew:                           0.008   Prob(JB):                       0.0486
Kurtosis:                       3.161   Cond. No.                     1.02e+03
==============================================================================
```

Treatment Effect:

- ATE: $5.32 per customer
- 95% CI: [4.59, 6.05]
- Standard Error: 0.371 (42% smaller than standard ATE)
- p-value < 0.001

Model Performance:

- R^2: 67.1% of revenue variance explained
- Substantial improvement from 1.1% in standard model
- Sample size: 5,556 customers
- Covariates: AOV, purchase frequency, recency, loyalty status

Key Insight: ML predictions (g_hat = 0.947) strongly capture baseline spending patterns

Implementing MLRATE

Metric	Standard ATE	ML ATE	Improvement
Treatment Effect	$5.10	$5.32	+$0.22
95% CI	[3.84, 6.36]	[4.59, 6.05]	42% narrower
CI Width	2.53	1.46	-1.07
Standard Error	0.644	0.371	42% smaller
R^2	1.1%	67.1%	+66.0%
p-value	<0.001	<0.001	Both significant

Key Insights:
✓ Both methods agree effect is ~$5 (validation!)
✓ MLRATE achieves 42% greater precision with same sample
✓ Variance explained jumped from 1.1% to 67.1%
✓ Future experiments can use 68% fewer customers for same power

Final Conclusions

Key Findings & Results

Experimental Results:

- Standard ATE: $5.10 increase ($p < 0.001$, 95% CI: [3.84, 6.36])
- MLRATE-Adjusted ATE: $5.32 ($p < 0.001$, 95% CI: [4.59, 6.05])
- Both estimates highly significant and consistent
- Treatment effect strongest among loyalty members and high-AOV customers

MLRATE Impact - The Power of Variance Reduction:

- Explained 67.1% of revenue variance (vs 1.1% in standard model)
- Reduced standard error by 42% (0.644 → 0.371)
- Narrowed confidence interval by 42% (width: 2.53 → 1.46)
- Achieved same statistical power with 68% fewer customers needed

Business Value:

- More precise estimates → more confident decisions
- Lower sample requirements → faster, cheaper experiments
- Demonstrates ML's value in causal inference

Business Recommendation

ROI Analysis:

- Gift cost: \$3.50 per customer
- Revenue increase: \$5.10 (Standard) / \$5.32 (MLRATE - more precise)
- Net profit: \$1.82 per customer (using MLRATE)
- Return on Investment: 52% ROI

Strategic Insights:

- Prioritize targeting loyalty members and high-value segments
- MLRATE enables faster, more cost-effective experimentation
- Future campaigns can test smaller effects with confidence

Next Steps: Scale rollout → Monitor retention → Test personalization

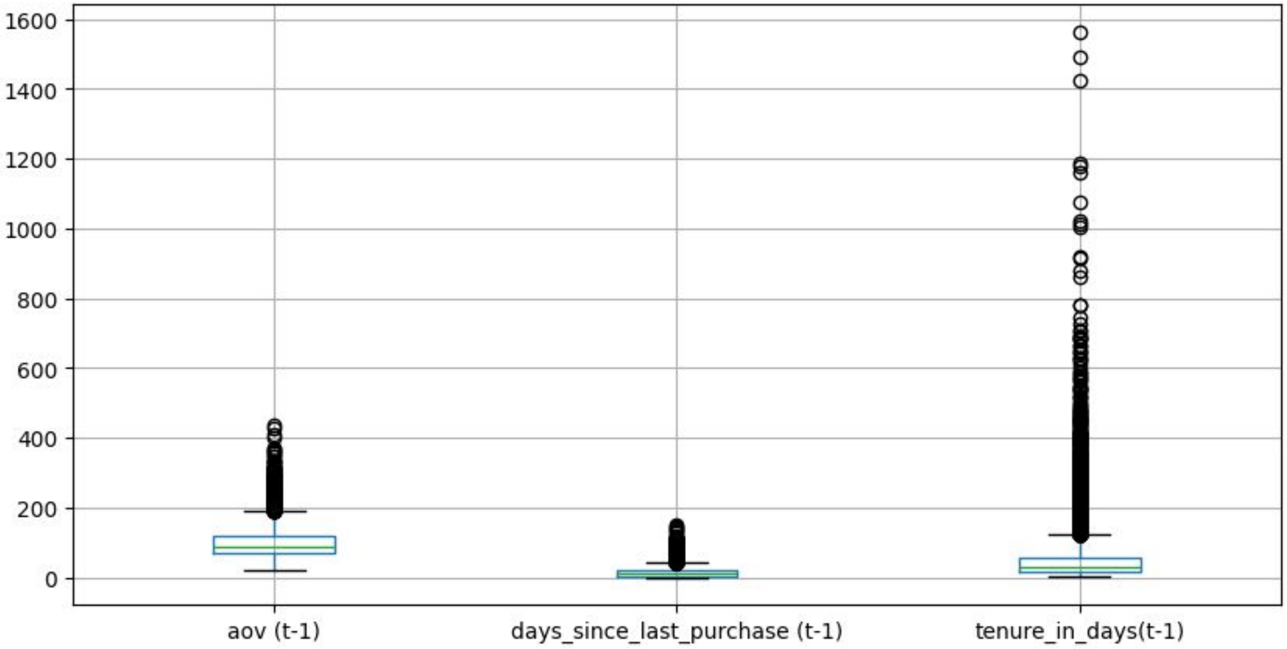
Thank you!

Questions?

Appendix

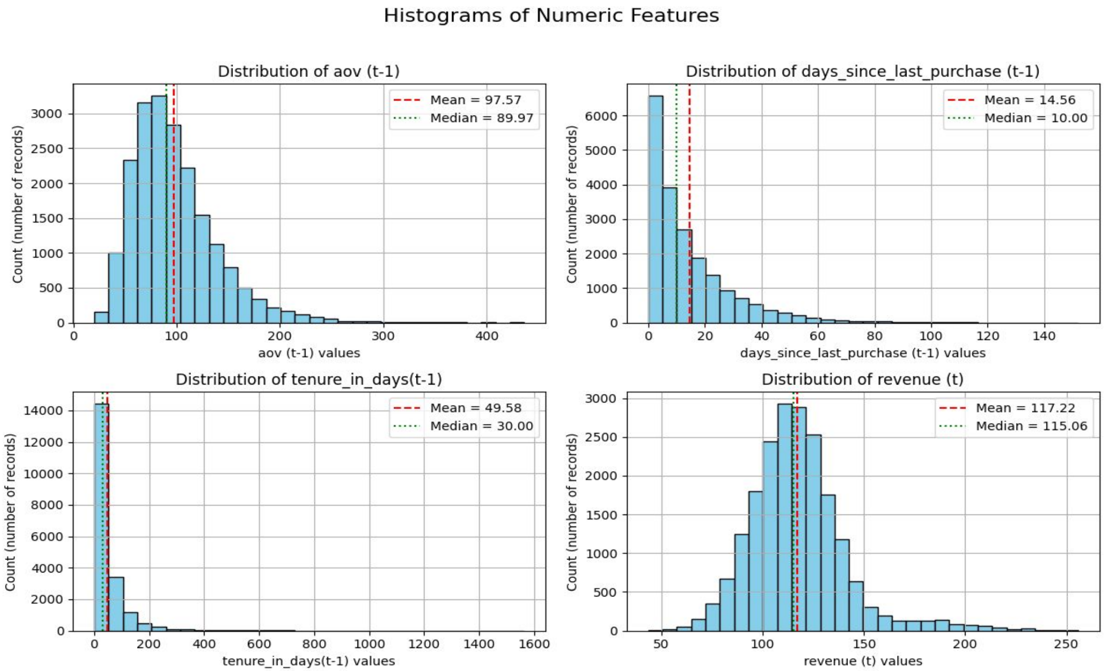
Pre-Experiment Data

Data Preprocessing - Checking for outliers



Pre-Experiment Data

EDA - Distribution of Numeric Features

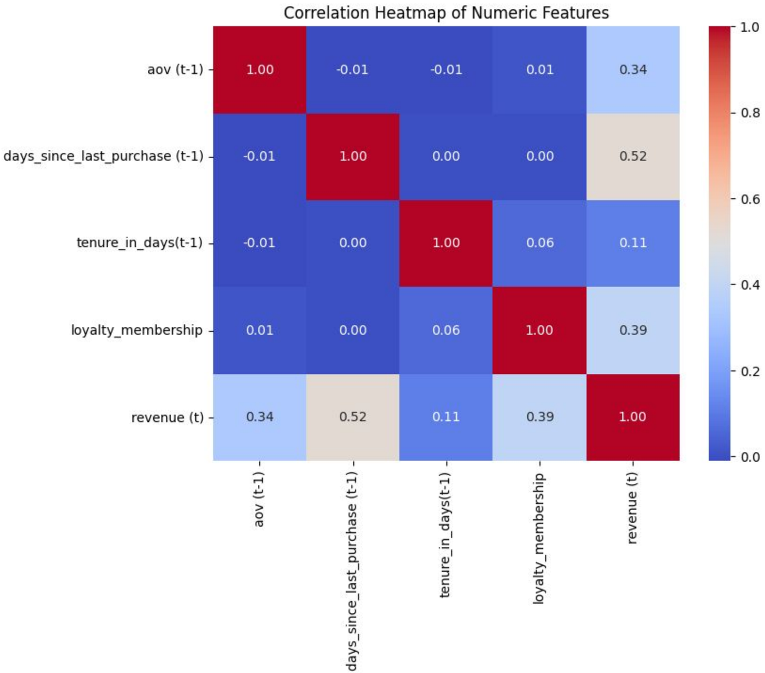


Pre-Experiment Data

EDA - Distribution of Loyalty Membership Features

Pre-Experiment Data

EDA - Relationships Between Numeric Features



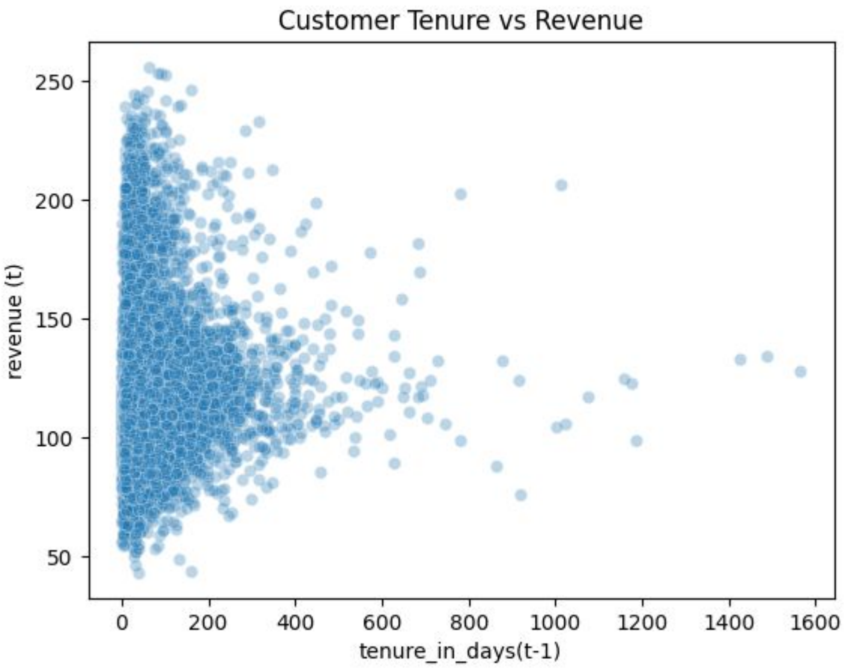
Pre-Experiment Data

EDA - Average Revenue and AOV by Loyalty Status

```
Average revenue by loyalty membership:
 loyalty_membership
0    113.354800
1    177.464888
Name: revenue (t), dtype: float64

Average AOV (t-1) by loyalty membership:
 loyalty_membership
0    97.445541
1    99.461549
Name: aov (t-1), dtype: float64
```

Pre-Experiment Data

EDA - Customer Tenure vs Revenue



Power Analysis

Method 1: Standard Power Analysis (T-Test)

```
from statsmodels.stats.power import TTestIndPower

mde_percent = 0.01
mean_revenue = df['revenue (t)'].mean()
diff = mde_percent * mean_revenue

analysis = TTestIndPower()
effect_size = diff / df['revenue (t)'].std()
sample_size = analysis.solve_power(
    effect_size=effect_size,
    power=0.9,
    alpha=0.05,
    alternative="two-sided",
)
print("Required sample size per group (T-Test):", round(sample_size))
```

```
Required sample size per group (T-Test): 8801
```

Power Analysis

Method 2: MLRATE Power Analysis

```
# Step 0: Define outcome and features
# y is our target (revenue at time t) that we want to explain
y = df['revenue (t)'].values

# x are the predictors (features from time t-1 and loyalty status)
# these will help us explain some of the variance in revenue
X = df[['aov (t-1)',
        'days_since_last_purchase (t-1)',
        'tenure_in_days(t-1)',
        'loyalty_membership']].values


# Step 1: Cross-fitting predictions
# we split the data into 2 folds so we can predict on data the model hasn't seen
kf = KFold(n_splits=2, shuffle=True, random_state=42)

# create empty array to store predictions
G = np.zeros_like(y)

# train on one half, predict on the other half
for train_idx, test_idx in kf.split(X):
    model = RandomForestRegressor(random_state=42)
    model.fit(X[train_idx], y[train_idx])        # train model
    G[test_idx] = model.predict(X[test_idx])     # save predictions
```

Power Analysis

Method 2: MLRATE Power Analysis

```
# Step 2: Fit OLS with G
# add a constant (intercept) to predictions
G = sm.add_constant(G)

# regress actual revenue on predicted revenue
# residuals here = the part of revenue we still can't explain
ols_model = sm.OLS(y, G).fit(cov_type="HC0")


# Step 3: Extract residuals & std
# get the unexplained part of revenue (residuals)
residuals = ols_model.resid

# calculate the standard deviation of residuals
resid_std = residuals.std(ddof=1)


# Step 4: Compute new effect size (MLRATE)
# define minimum detectable effect (mde) as 1% of average revenue
mde_percent = 0.01
mean_revenue = df['revenue (t)'].mean()
diff = mde_percent * mean_revenue

# effect size = mde / std of residuals
effect_size_mlr = diff / resid_std
```

Power Analysis

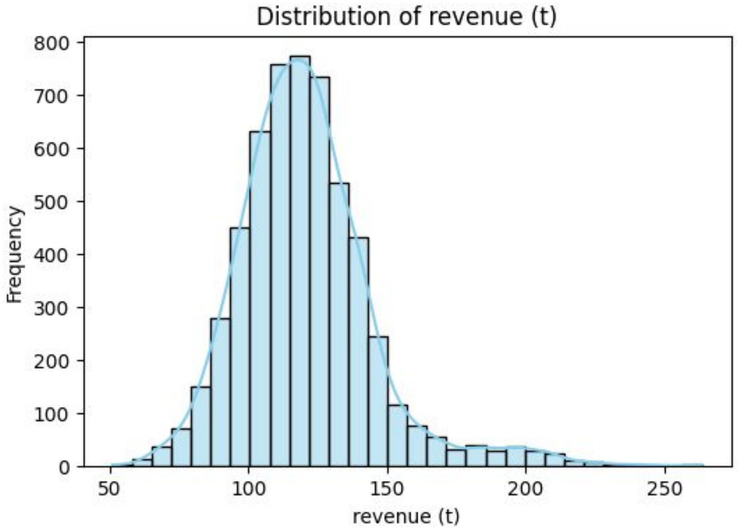
Method 2: MLRATE Power Analysis

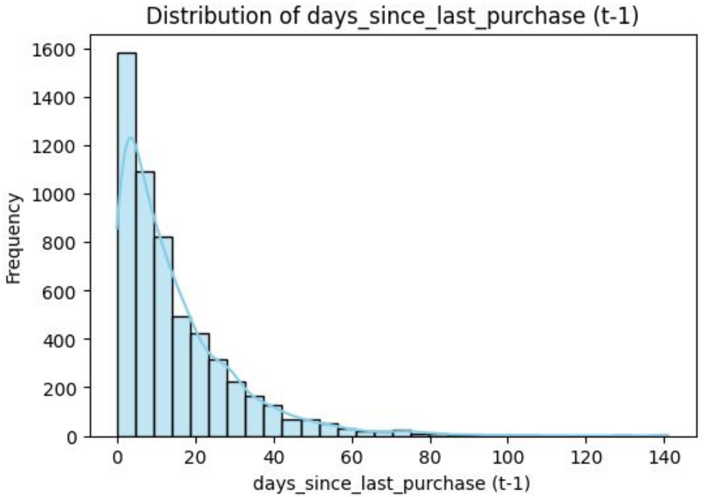
```
# Step 5: Power analysis with new effect size
# calculate required sample size per group (90% power, 5% alpha, two-sided test)
analysis = TTestIndPower()
sample_size_mlr = analysis.solve_power(
    effect_size=effect_size_mlr,
    power=0.9,
    alpha=0.05,
    alternative="two-sided",
)

print("Required sample size per group (MLRATE):", round(sample_size_mlr))
```

```
Required sample size per group (MLRATE): 2778
```

Experiment Data

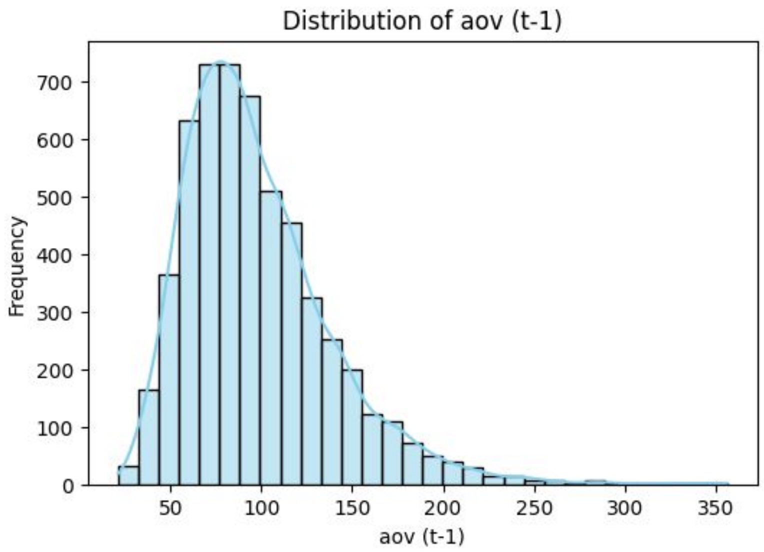
EDA - Histograms (Distribution of each variable)

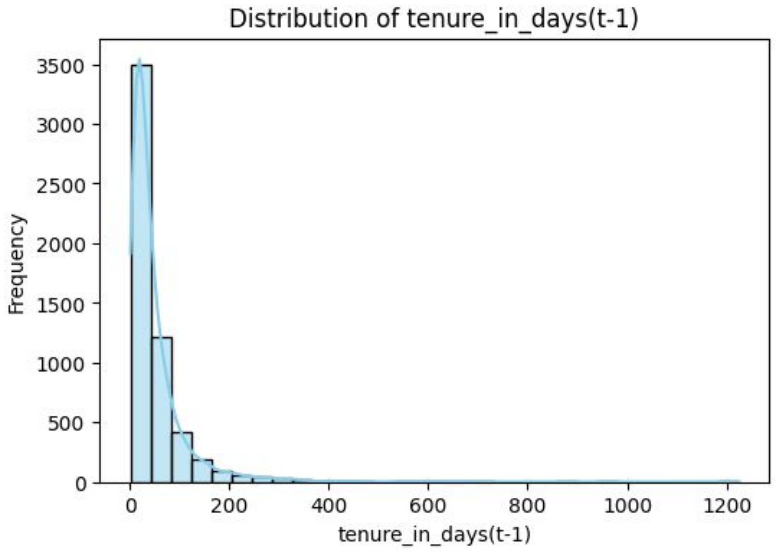




Experiment Data

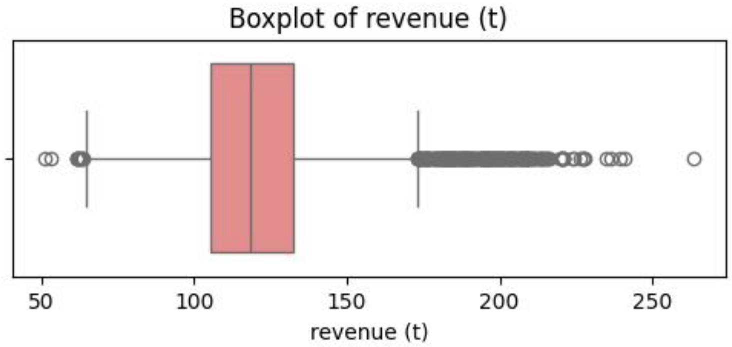
EDA - Histograms (Distribution of each variable)

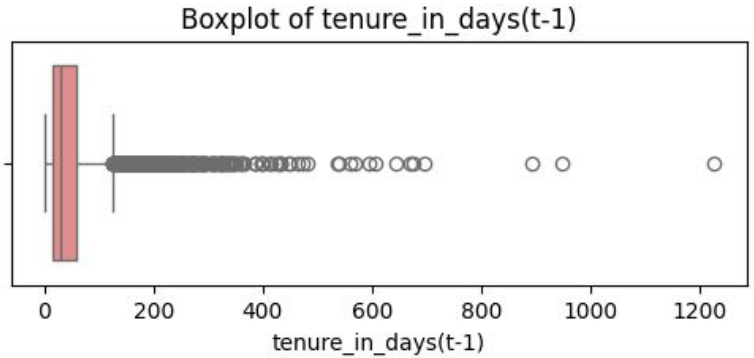


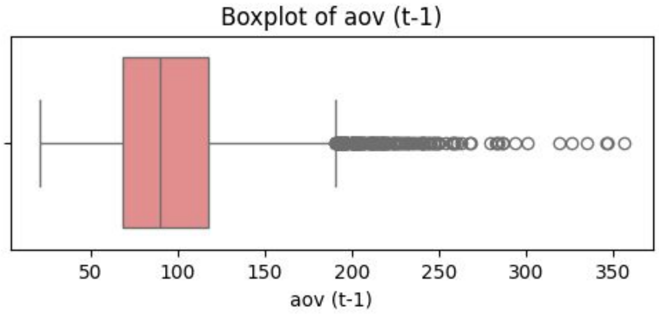


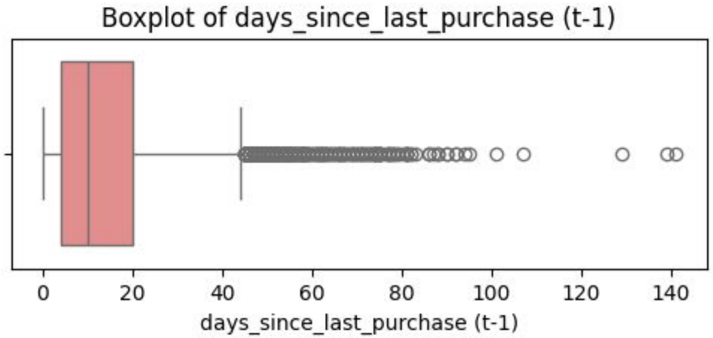
Experiment Data

EDA - Step 3: Boxplots (Outlier detection)









Estimating the ATE

```
# 1. define treatment and outcome variables
assignment = df["assignment"]          # 1 = treated, 0 = control
revenue = df["revenue (t)"]            # post-experiment revenue

# 2. add a constant term (intercept)
X = sm.add_constant(assignment)
y = revenue

# 3. fit ols regression with robust standard errors
model = sm.OLS(y, X).fit(cov_type="HC0")

# 4. print regression summary
print(model.summary(xname=["const", "treatment (gift sent)"]))

# 5. extract results
ate = model.params["assignment"]
ci_lower, ci_upper = model.conf_int().loc["assignment"]
ci_width = ci_upper - ci_lower
p_value = model.pvalues["assignment"]

print(f"\nestimated ate: ${ate:.2f}")
print(f"95% ci: [{ci_lower:.2f}, {ci_upper:.2f}] (width = {ci_width:.2f})")
print(f"p-value: {p_value:.4f}")
```

Implementing MLRATE

```
# Step 0. Rename treatment column for consistency
df.rename(columns={'Group': 'treatment'}, inplace=True)

# Convert treatment to numeric (0 = Control, 1 = Treatment)
df['treatment'] = df['treatment'].map({'Control': 0, 'Treatment': 1})

#  Step 1. Define function for out-of-fold predictions
def out_of_fold_predict_baseline(X, y, model=None, n_splits=5):
    if model is None:
        model = RandomForestRegressor(n_estimators=200, random_state=42)

    oof_preds = np.zeros(len(X))
    kf = KFold(n_splits=n_splits, shuffle=True, random_state=42)

    for train_idx, val_idx in kf.split(X):
        X_train, X_val = X.iloc[train_idx], X.iloc[val_idx]
        y_train = y.iloc[train_idx]

        model.fit(X_train, y_train)
        oof_preds[val_idx] = model.predict(X_val)

    return oof_preds
```

Implementing MLRATE

```
#  Step 2. Predict baseline (G(x))
T = df['treatment']  # treatment assignment (0 or 1)
y = df['revenue (t)']  # observed post-treatment outcome
X = df[['aov (t-1)', 'days_since_last_purchase (t-1)', 'tenure_in_days(t-1)', 'loyalty_membership']]  # pre-treatment features

g_hat = out_of_fold_predict_baseline(X, y)

#  Step 3. Center G(x)
g_bar = g_hat.mean()
g_centered = g_hat - g_bar

#  Step 4. Build regression design
X_reg_df = pd.DataFrame({
    'T': T,
    'g_hat': g_hat,
    'T * (g_hat - g_bar)': T * g_centered
})
X_reg_df = sm.add_constant(X_reg_df)
```

Implementing MLRATE

```python
#  Step 5. Fit MLRATE model
ols = sm.OLS(y, X_reg_df).fit(cov_type="HC0")

#  Step 6. Extract ATE results
ate = ols.params['T']
ci_lower, ci_upper = ols.conf_int().loc['T']

print(f"MLRATE ATE: {ate:.4f}")
print(f"95% CI: [{ci_lower:.4f}, {ci_upper:.4f}]")
print(f"p-value: {ols.pvalues[1]:.8f}")

print("\nFeature matrix shape:", X.shape)
print("\nFirst few outcomes:\n", y.head())
print("\nTreatment group counts:\n", T.value_counts())

print("\nModel Summary:\n", ols.summary())
```

Estée Lauder 1B

Unwrapping Customer Delight: Optimizing Surprise Gift Strategies with Randomized Controlled Trials and MLRATE

DECEMBER 2025

AI Studio Final Project

BREAK THROUGH TECH

Meet Our Team

Sehr Abrar
CUNY: City College

Lydia Aubourg
Rutgers University

Ava Leung
Rutgers University

Xiaoyan Li
CUNY: Hunter College

Nidhi Parvathala
UT Dallas

Charitha Sarraju
UT Dallas

Sandy Wu
Stony Brook University

To measure the impact of surprise gifts on customer spending using machine learning and provide insights for marketing ROI.