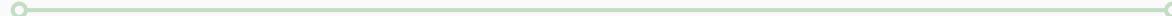




# Milestone #:6

## Implementing MLRATE



Estee Lauder 1B



01

# Code Implementation

## Implementing MLRATE

# Code Implementation

```
# 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
```

# Code Implementation

```
# 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)
```

# Code Implementation

```
# 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())
```

# Code Results

```
MLRATE ATE: 5.3194
95% CI: [4.5915, 6.0474]
p-value: 0.00000000

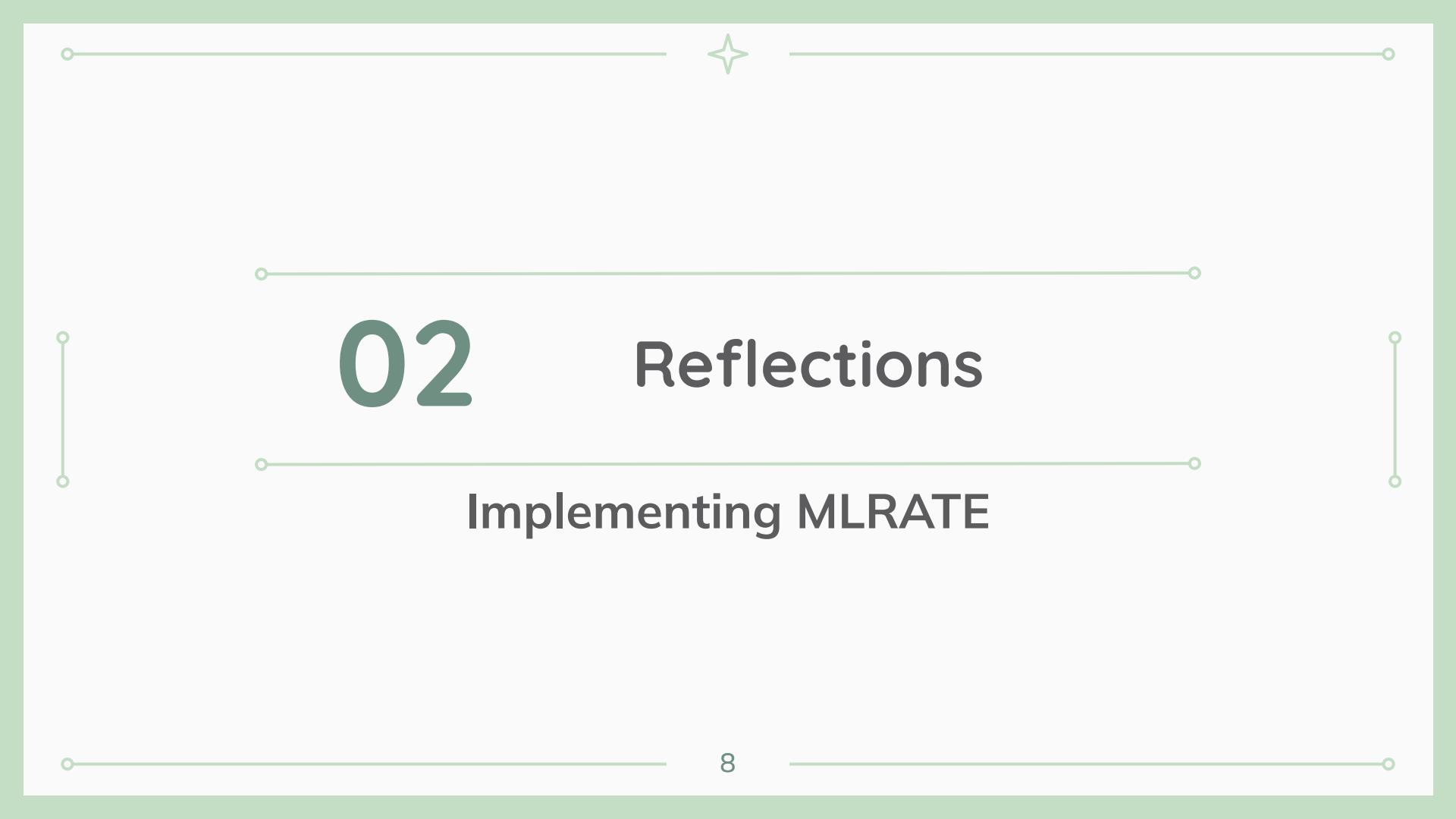
Feature matrix shape: (5556, 4)

First few outcomes:
 9841    140.9200
 412     155.2110
 8337    129.8745
 5118    133.7800
 9479    101.6000
Name: revenue (t), dtype: float64

Treatment group counts:
 treatment
 1      2783
 0      2773
Name: count, dtype: int64
```

# Code Results

```
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
=====
```



02

Reflections

Implementing MLRATE

# Comparing Unadjusted ATE vs. MLRATE ATE

## Unadjusted ATE vs. MLRATE ATE:

- Unadjusted ATE: 5.10
- MLRATE ATE: 5.3194

## 95% Confidence Interval:

- Unadjusted: [3.838, 6.365]
  - Unadjusted CI Width:  $6.365 - 3.838 = 2.527$
- MLRATE: [4.5915, 6.0474]
  - MLRATE CI Width:  $6.0474 - 4.5915 = 1.4559$
- MLRATE → slightly narrower

## Interpretation:

- Adding the predictive baseline  $G(x)$  improved precision (narrowed CI).
- The treatment effect remains statistically significant ( $p \approx 0$ ).
- Higher precision increases confidence in the decision to implement the gift strategy.

## Baseline Model $G(x)$ Predictiveness & Model Fit

### Predictiveness of $G(x)$ :

- Coefficient for  $g_{\hat{}}: 0.9470, p < 0.001 \rightarrow$  very predictive
- $R^2 = 0.671 \rightarrow G(x)$  explains ~67% of revenue variance

### Interaction term $T * (g_{\hat{}} - g_{\bar{}})$ :

- Coefficient  $\approx -0.003, p = 0.871 \rightarrow$  no significant treatment heterogeneity

### Implications:

- Baseline is strong  $\rightarrow$  MLRATE gives meaningful improvement in CI
- If  $G(x)$  were weak, MLRATE would offer little precision gain

# Reflection & Business Takeaways

## Precision & decision-making:

- Narrower CI increases confidence that treatment effect is positive
- Supports “launch gift strategy” decision

## Coefficient interpretation:

- $\text{const} = 3.6446 \rightarrow$  baseline revenue for average customer without treatment
- $T = 5.3194 \rightarrow$  estimated lift from treatment
- $g_{\hat{}} = 0.9470 \rightarrow$  strong predictor of revenue
- $T * (g_{\hat{}} - g_{\bar{}}) \approx 0 \rightarrow$  treatment effect consistent across predicted revenue

## Model reliability:

- No signs of leakage  $\rightarrow$  out-of-fold predictions used
- Condition number  $= 1.02e3 \rightarrow$  slight multicollinearity, but not critical



**Thank you!**

Notebook Link