📊 Comprehensive Comparative Study of Uplift / True Lift Modeling Approaches

---

🔑 Common Notations

- **X**: Feature vector (customer demographics, transaction history, etc.)
- **Y**: Outcome (binary: 1 = responded, 0 = not responded)
- **T**: Treatment indicator (1 = received campaign, 0 = control)
- **f, f_1, f_0**: Predictive models
- **ê(X)**: Propensity score = P(T=1 | X)
- **τ(X)**: Uplift / Individual Treatment Effect (ITE)
- **m̂(X)**: Baseline expected outcome

---

1. S-Learner (Single Model)

Idea / Intuition
- Train one model with treatment flag as a feature.
- Predict outcomes under both treated & control conditions.

Pros
- Simple, uses all data.
- Good baseline.

Cons
- Blurs treatment vs. control heterogeneity.
- Biased if treatment groups differ a lot.

Equation
$\tau(X) = f(X,1) - f(X,0)$

Workflow
1. Add T to features.
2. Train single model.
3. Predict f(X,1) and f(X,0).
4. Subtract for uplift.

---

2. T-Learner (Two Models)

Idea / Intuition

- Build separate models for treated and control.

Pros
- Captures heterogeneity.
- Intuitive “what-if” logic.

Cons
- Needs large balanced data.
- Errors from both models add up.

Equation
$\tau(X) = f_1(X) – f_0(X)$

Workflow
1. Split into treated & control.
2. Train f_1, f_0 separately.
3. Predict both, subtract.

---

3. Class Transformation

Idea / Intuition
- Re-label persuadables into a binary class problem.

Pros
- Uses standard classifiers.
- Focuses directly on uplift.

Cons
- Labels noisy.
- Smaller effective dataset.

Equation
Z = 1 if (T=1 & Y=1) or (T=0 & Y=0), else 0

Workflow
1. Create Z labels.
2. Train classifier g(X).
3. Use g(X) as uplift score.

---

4. U-Learner

Idea / Intuition
- Residual-based using baseline & propensity.

Pros
- Efficient for small samples.
- Balances outcome & treatment.

Cons
- Sensitive to poor propensity models.

Equation
$H = (Y - \hat{m}(X)) / (T - \hat{e}(X))$
$\tau(X) = E[H \mid X]$

Workflow
1. Fit $\hat{m}(X)$, $\hat{e}(X)$.
2. Compute H.
3. Regress H on X.

---

5. X-Learner

Idea / Intuition
- Handles imbalanced groups by imputing counterfactuals.

Pros
- Works well with imbalance.
- Reduces variance vs. T-Learner.

Cons
- Multi-step, more complex.

Equation
$D^1 = Y^1 - f_0(X^1)$
$D^0 = f_1(X^0) - Y^0$
$\tau(X) = g(X) \cdot h_0(X) + (1 - g(X)) \cdot h_1(X)$

Workflow
1. Train f_1, f_0.
2. Impute D^1, D^0.
3. Train h_1, h_0.
4. Combine with g(X).

---

6. R-Learner

Idea / Intuition
- Orthogonalizes treatment effect from outcome.

Pros
- Double-robust.
- Handles observational data.

Cons
- Needs strong $\hat{m}$ and $\hat{e}$ models.

Equation
$\hat{Y} = Y - \hat{m}(X)$
$\tilde{T} = T - \hat{e}(X)$
$\hat{Y} = \tilde{T} \cdot \tau(X) + \varepsilon$

Workflow
1. Estimate $\hat{m}(X)$, $\hat{e}(X)$.
2. Compute residuals.
3. Regress residuals.

---

7. Uplift Trees & Forests

Idea / Intuition
- Decision trees split to maximize uplift gain.

Pros
- Very interpretable.
- Produces clear rules.

Cons
- Risk of overfitting.
- Less stable than ensembles.

Workflow
1. Grow trees optimizing uplift.
2. Leaves estimate treatment effect.
3. Combine trees (forests) for stability.

---

8. Causal Forests

Idea / Intuition
- Ensemble of uplift trees for heterogeneous effects.

Pros
- Captures complex heterogeneity.
- State-of-the-art for causal ML.

Cons
- Computationally heavy.
- Black-box for business users.

Workflow
1. Build many uplift trees.
2. Aggregate results.
3. Estimate $\tau(X)$ across groups.

—