COHORT REVENUE & RETENTION ANALYSIS: A BAYESIAN APPROACH

JUAN CAMILO ORDUZ

Abstract. We present a bayesian approach to model cohort-level retention rates and revenue over time. We use bayesian additive regression trees (BART) to model the retention component which we couple with a linear model to model the revenue component. This method is flexible enough to allow adding additional covariates to both model components. This bayesian model allows us to quantify the uncertainty in the estimation, understand the effect of the covariates on the retention through partial dependence plots (PDP), individual conditional expectation (ICE) plots and last but not least forecast the future revenue and retention rates.

Contents

1. Introduction

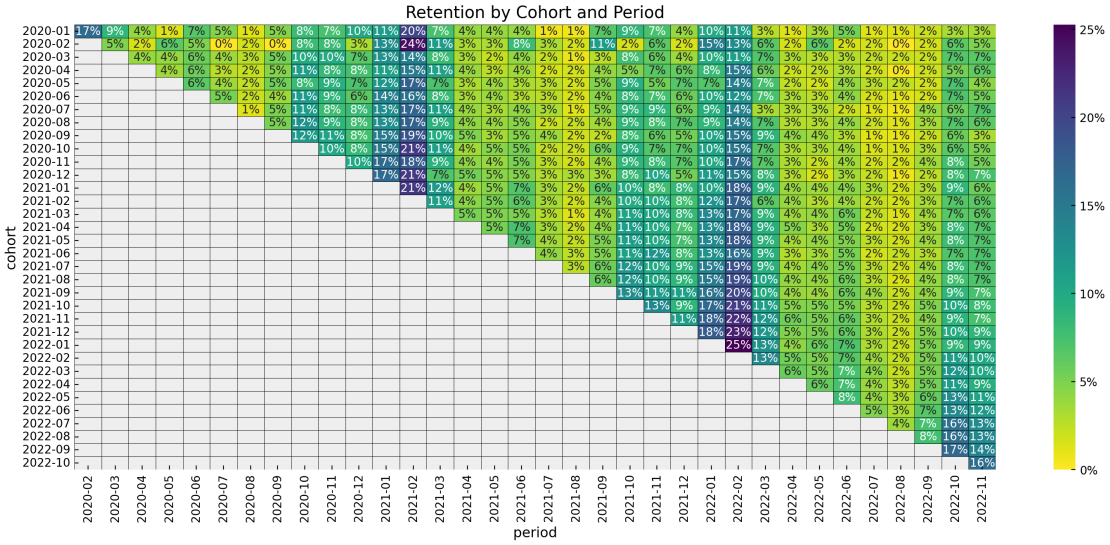
Retention and customer lifetime value estimation are one of the most important aspects to understand customer behavior. There are many ways to model retention and revenue by modeling individual-level purchase behavior for both the contractual and the non-contractual setting, see for example the work by Fader and Herdie [2] and [1] respectively[1]. In real cases, one is interested in modeling retention and revenue at cohort-level. There are (at least) three options to use the techniques mentioned above to model cohort-level retention:

(1) Pool the cohorts together and model the retention and revenue as a whole.
(2) Un-pool the cohorts and model each cohort separately.
(3) Try to model the cohorts jointly.

See [3] for more details on these approaches. One of the limitations of those approaches is that they are not flexible enough to model seasonality and add external regressors. One

Date: February 26, 2023.

[1]Our definition of retention is what they call survival curve. See precise definitions below.



can argue that seasonality is not that important when trying to estimate the customer-lifetime-value. Nevertheless, in practice, there are business models on which the customer base is very seasonal.

In this work, we present a bayesian approach to model cohort-level retention rates from a top-down perspective. We do not model the individual-level purchase behavior but rather the retention and revenue at cohort matrices. This approach allows for modeling non-linear relationships between cohorts, adding seasonality and external regressors. Concretely, we use bayesian additive regression trees (BART, see [7]) to model the retention component and we couple it with a linear model to model the revenue.

$$
\begin{aligned}
\text{Revenue} &\sim \text{Gamma}(N_{\text{active}}, \lambda) \\
\log(\lambda) &= (\text{intercept} \\
&\quad + \beta_{\text{cohort age}}\text{cohort age} \\
&\quad + \beta_{\text{age}}\text{age} \\
&\quad + \beta_{\text{cohort age}\times\text{age}}\text{cohort age} \times \text{age}) \\
N_{\text{active}} &\sim \text{Binomial}(N_{\text{total}}, p) \\
\text{logit}(p) &= \text{BART}(\text{cohort age}, \text{age}, \text{month})
\end{aligned}
$$

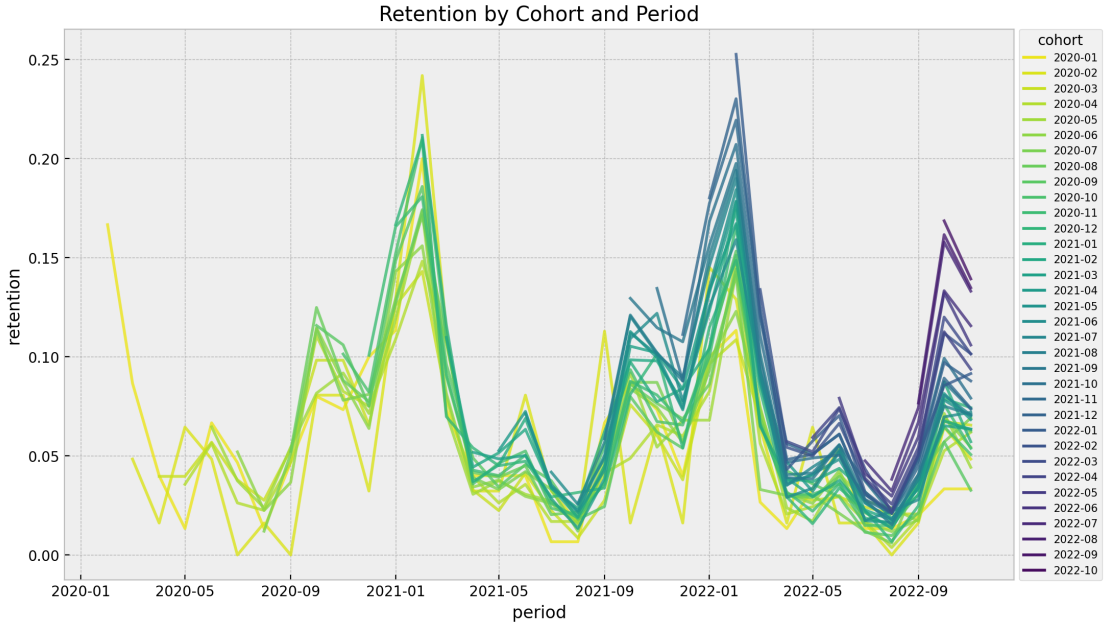
This work is the result of a sequence of blog posts where all the details on the code and implementation are presented, see [4], [5] and [6].

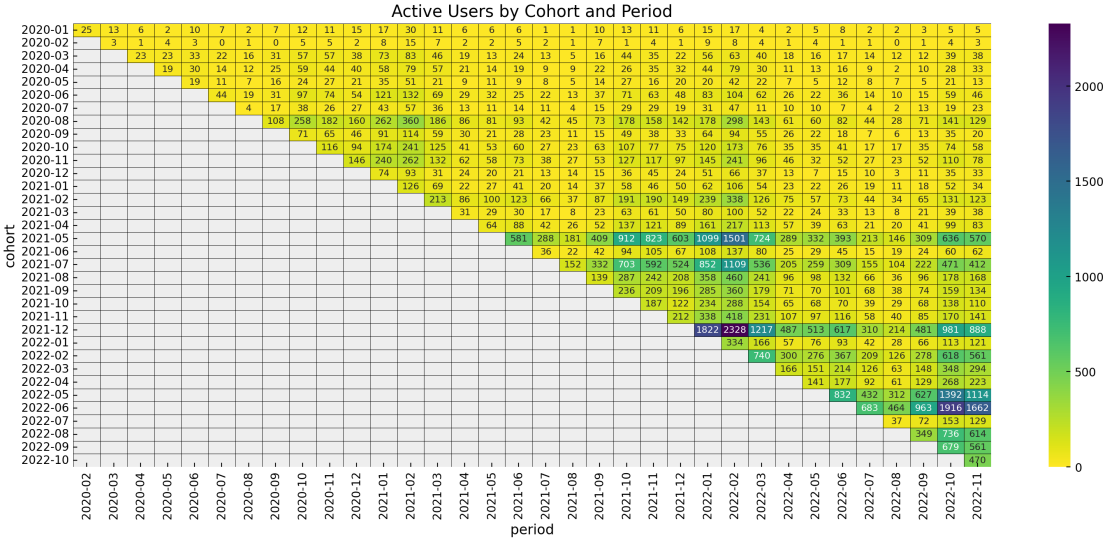
2. Data and Exploratory

3. Model Specification and Diagnostics

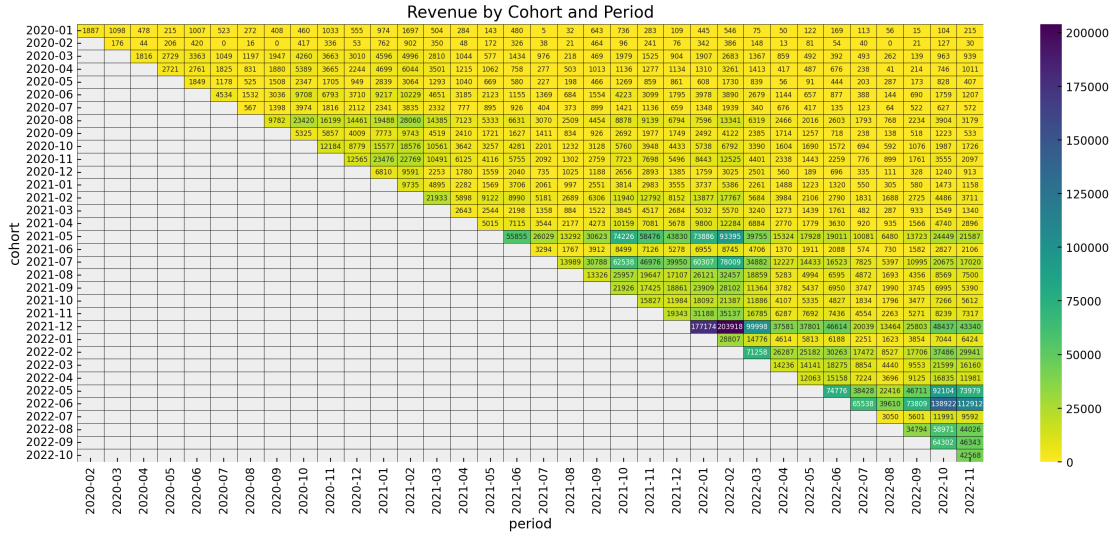
4. Predictions

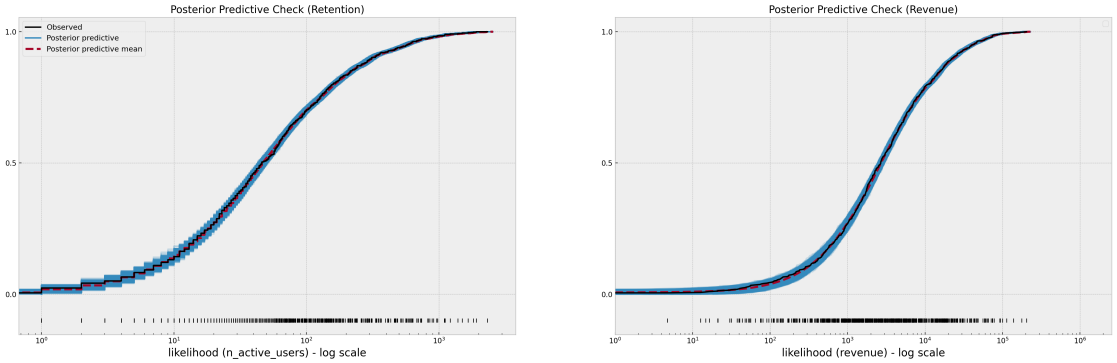
4.1. **In-Sample Predictions.**





4.2. **Out-of-Sample Predictions.**





Appendix: Python Code

LISTING 1. Python example

```python
import pymc_bart as pmb
import pymc as pm


with pm.Model(coords={"feature": features}) as model:

    # --- Data ---
    model.add_coord(name="obs", values=train_obs_idx, mutable=True)
    age_scaled = pm.MutableData(
        name="age_scaled", value=train_age_scaled, dims="obs"
    )
    cohort_age_scaled = pm.MutableData(
        name="cohort_age_scaled", value=train_cohort_age_scaled, dims="obs"
    )
    x = pm.MutableData(name="x", value=x_train, dims=("obs", "feature"))
```

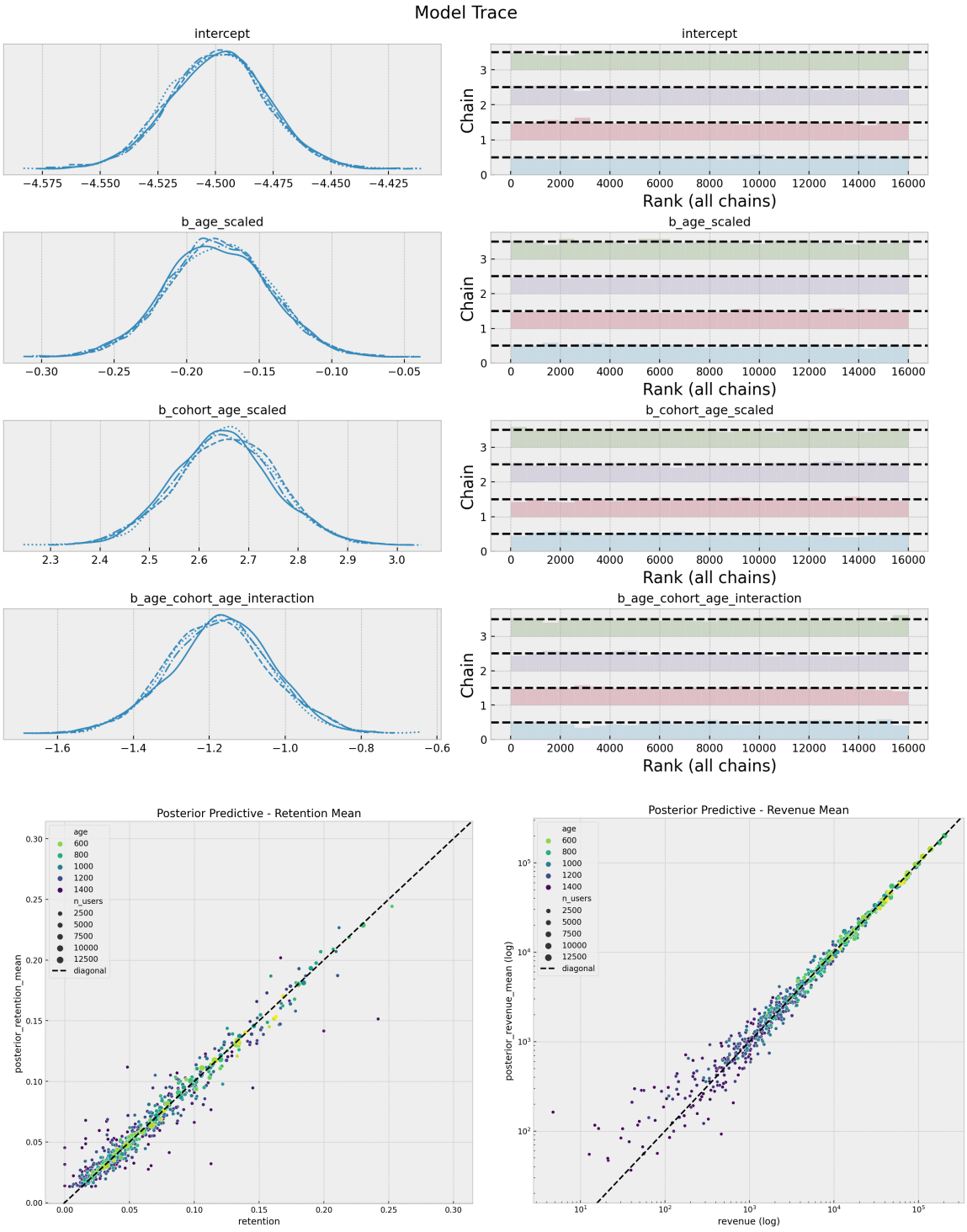


FIGURE 1. My caption

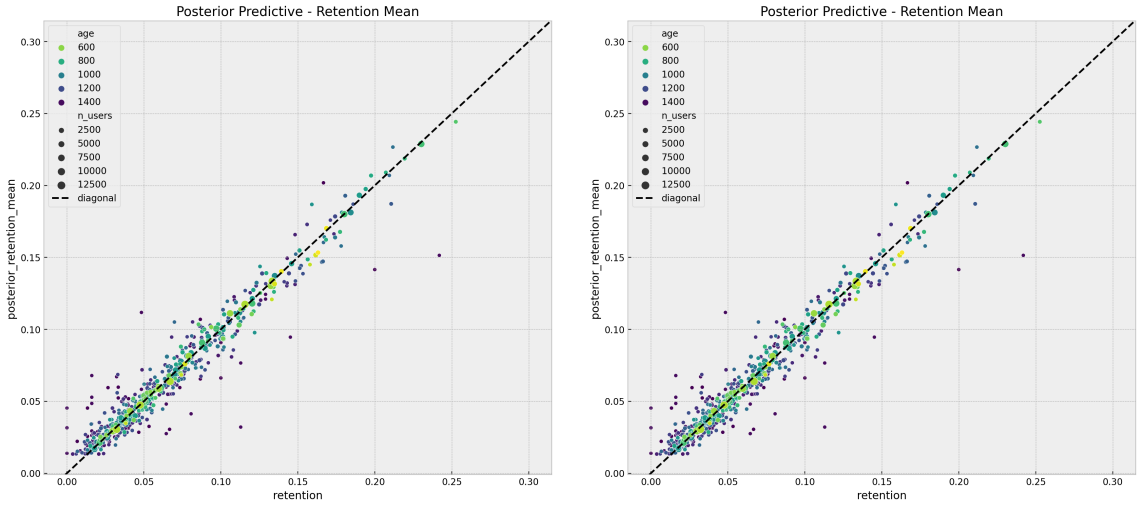
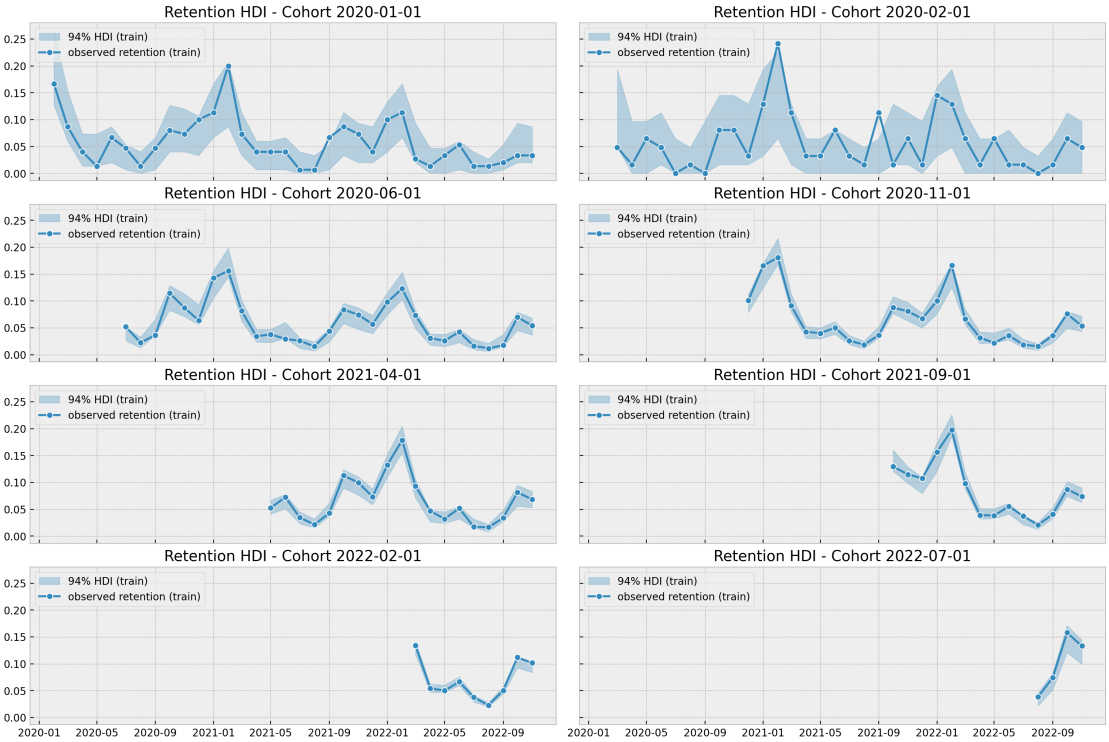
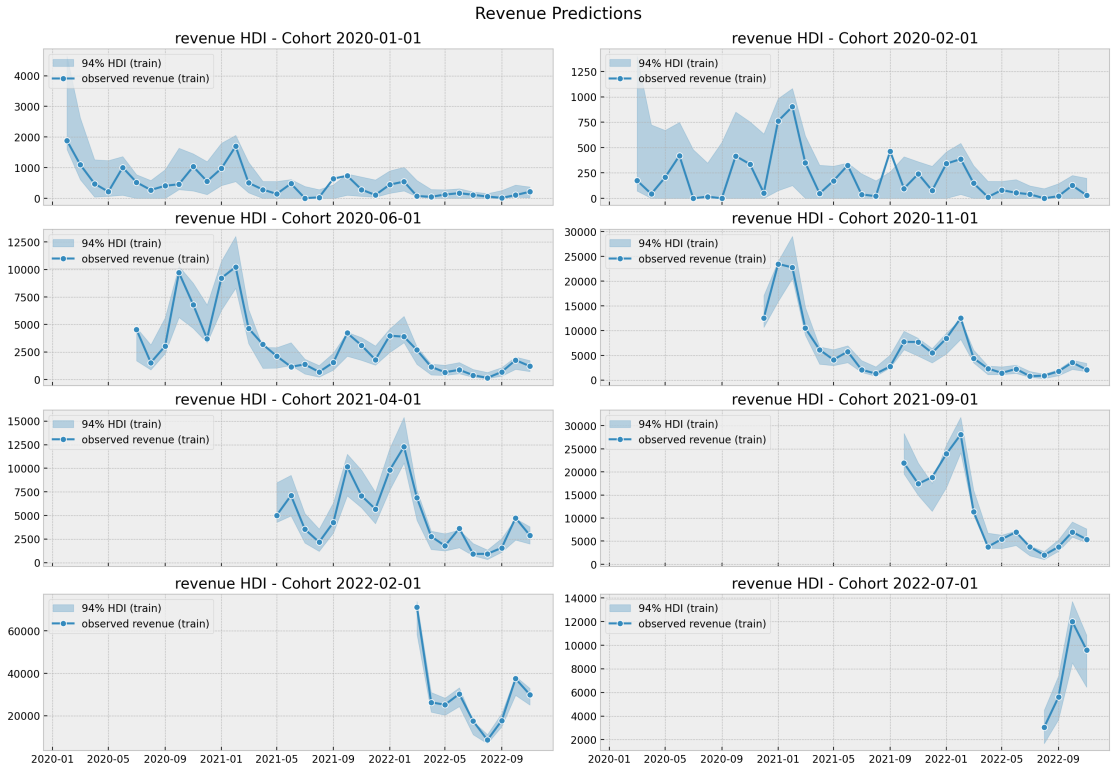


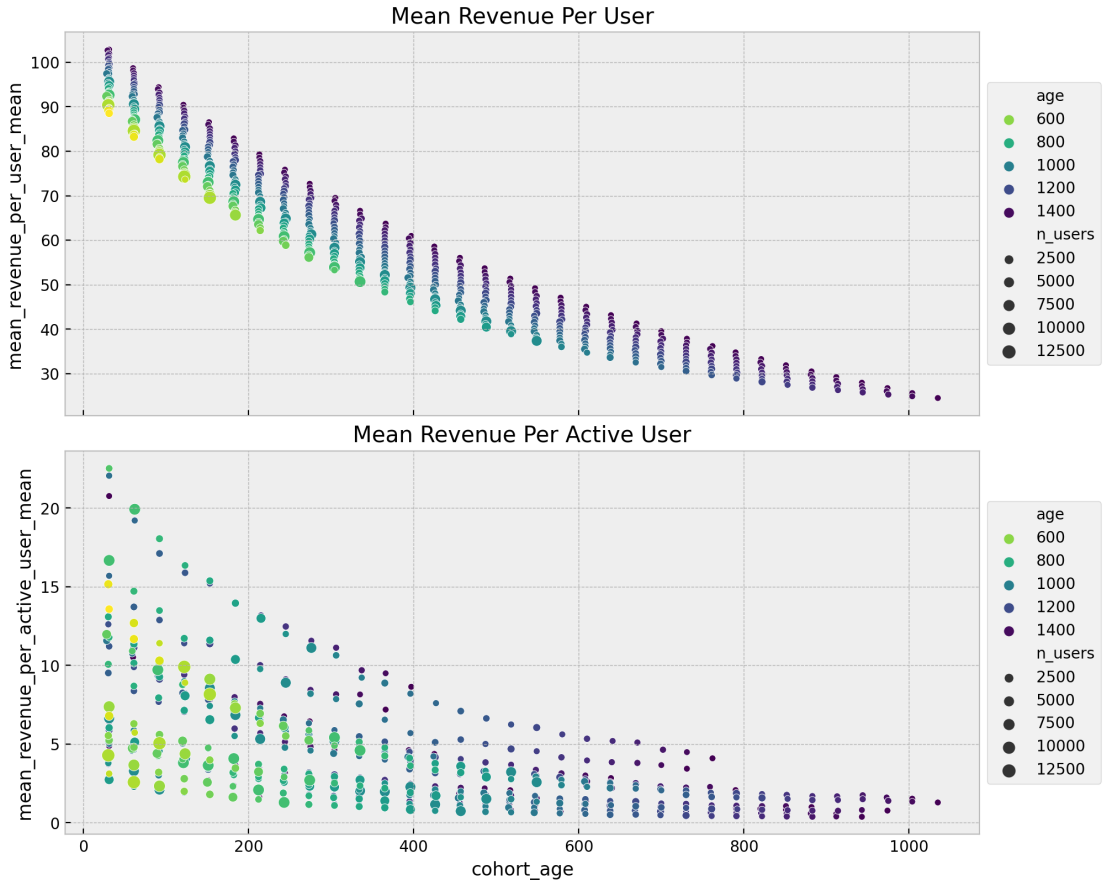
FIGURE 2. My caption



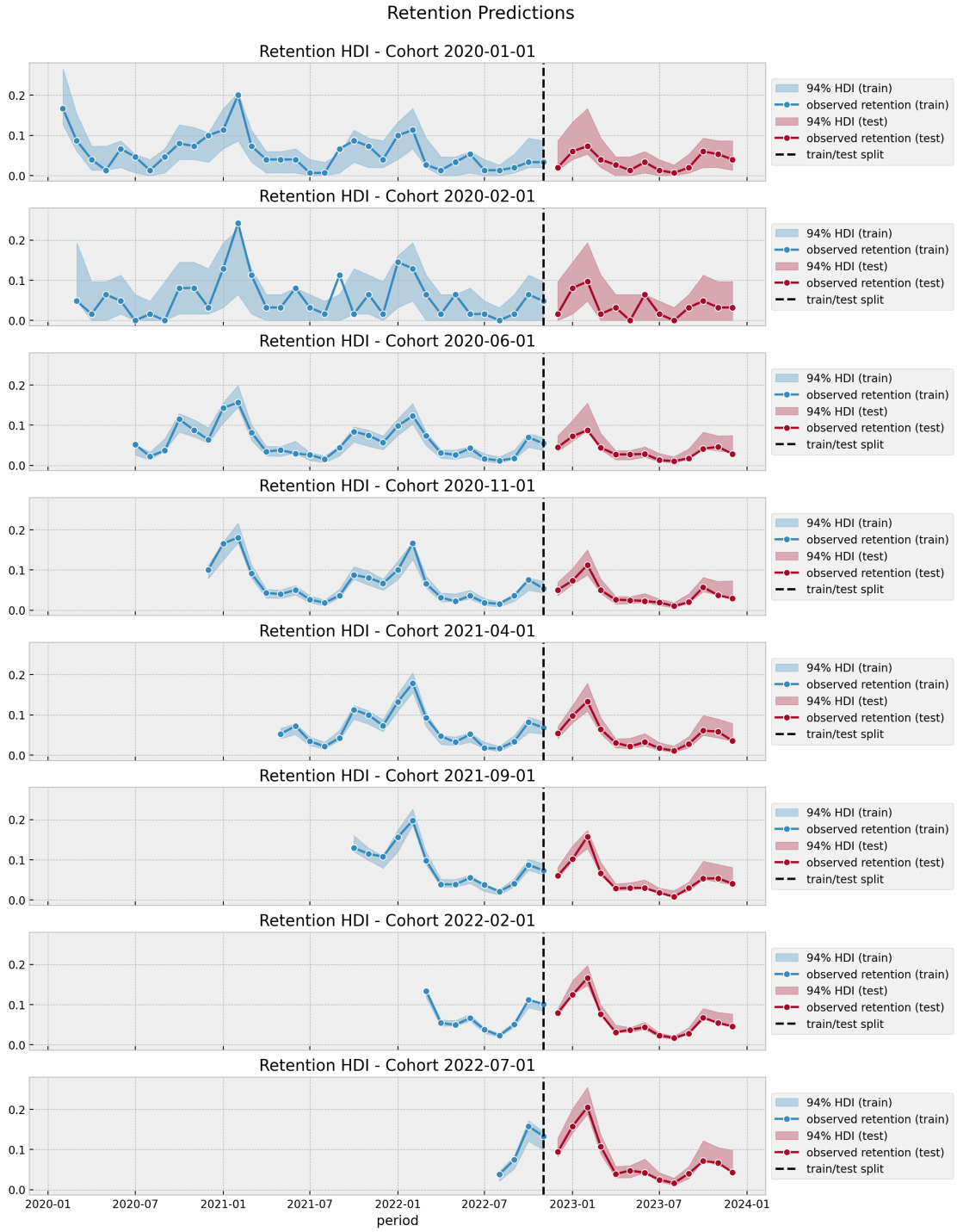

```
    n_users = pm.MutableData(name="n_users", value=train_n_users, dims="obs")
    n_active_users = pm.MutableData(
        name="n_active_users", value=train_n_active_users, dims="obs"
    )
    revenue = pm.MutableData(name="revenue", value=train_revenue, dims="obs")
```

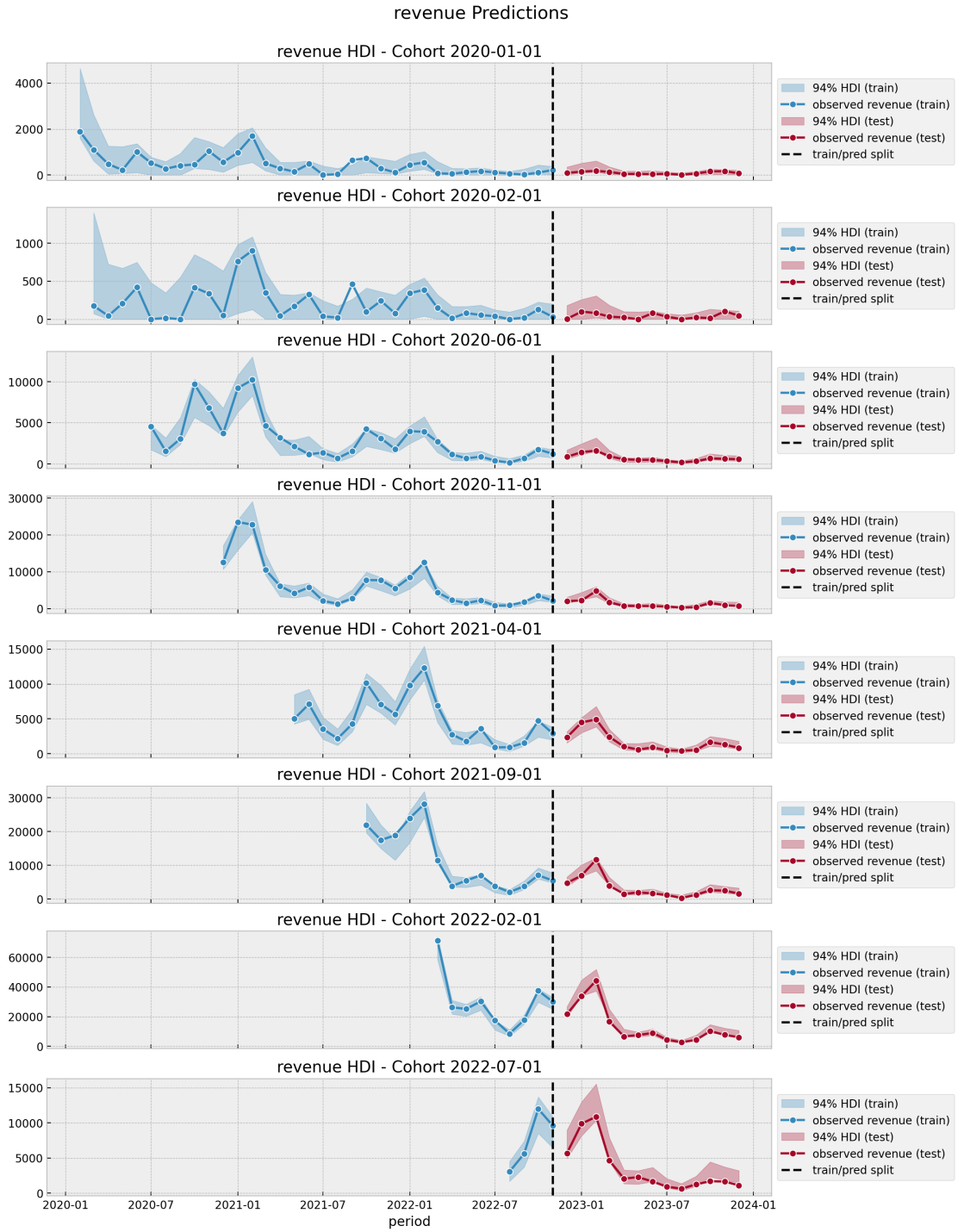



```
    # --- Priors ---
    intercept = pm.Normal(name="intercept", mu=0, sigma=1)
    b_age_scaled = pm.Normal(name="b_age_scaled", mu=0, sigma=1)
    b_cohort_age_scaled = pm.Normal(name="b_cohort_age_scaled", mu=0, sigma=1)
    b_age_cohort_age_interaction = pm.Normal(
        name="b_age_cohort_age_interaction", mu=0, sigma=1
    )

    # --- Parametrization ---
    # The BART component models the image of the retention rate under the
    # logit transform so that the range is not constrained to [0, 1].
    mu = pmb.BART(name="mu", X=x, Y=train_retention_logit, m=50, dims="obs")
    # We use the inverse logit transform to get the retention rate
    # back into [0, 1].
    p = pm.Deterministic(name="p", var=pm.math.invlogit(mu), dims="obs")
    # We add a small epsilon to avoid numerical issues.
    p = pt.switch(pt.eq(p, 0), eps, p)
    p = pt.switch(pt.eq(p, 1), 1 - eps, p)

    # For the revenue component we use a Gamma distribution where we
    # combine the number of estimated active users with the average
    # revenue per user.
```

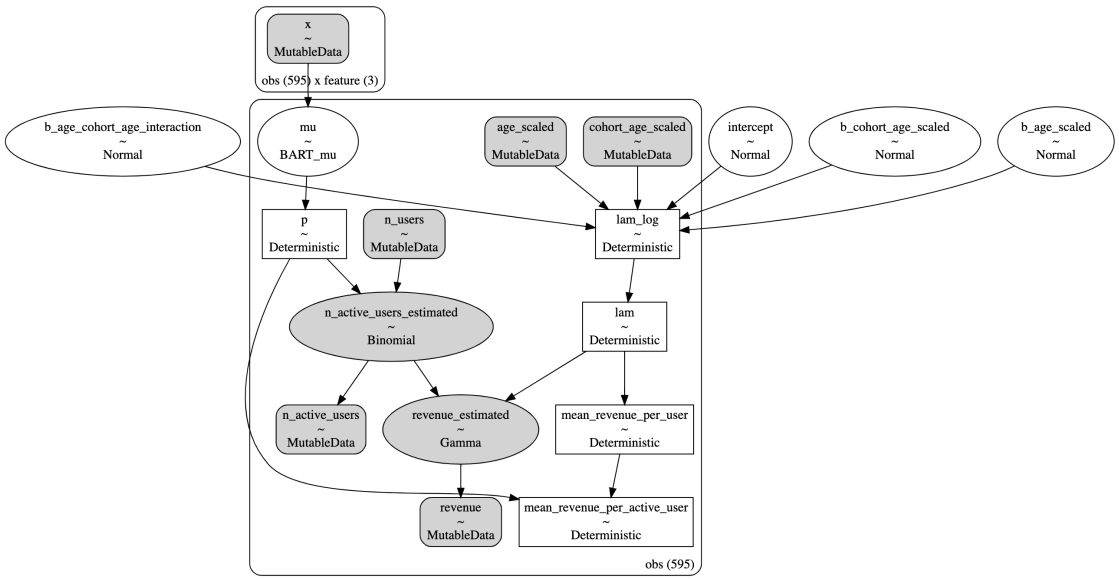



```python
    lam_log = pm.Deterministic(
        name="lam_log",
        var=intercept
        + b_age_scaled * age_scaled
        + b_cohort_age_scaled * cohort_age_scaled
        + b_age_cohort_age_interaction * age_scaled * cohort_age_scaled,
        dims="obs",
    )

    lam = pm.Deterministic(name="lam", var=pm.math.exp(lam_log), dims="obs")

    # --- Likelihood ---
    n_active_users_estimated = pm.Binomial(
        name="n_active_users_estimated",
        n=n_users,
        p=p,
        observed=n_active_users,
        dims="obs",
    )
```


Retention Predictions
Retention HDI - Cohort 2020-01-01
Retention HDI - Cohort 2020-02-01
Retention HDI - Cohort 2020-06-01
Retention HDI - Cohort 2020-11-01
Retention HDI - Cohort 2021-04-01
Retention HDI - Cohort 2021-09-01
Retention HDI - Cohort 2022-02-01
Retention HDI - Cohort 2022-07-01
94% HDI (train)
observed retention (train)
94% HDI (test)
observed retention (test)
train/test split
0.0
0.1
0.2
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01
period


revenue Predictions
revenue HDI - Cohort 2020-01-01
revenue HDI - Cohort 2020-02-01
revenue HDI - Cohort 2020-06-01
revenue HDI - Cohort 2020-11-01
revenue HDI - Cohort 2021-04-01
revenue HDI - Cohort 2021-09-01
revenue HDI - Cohort 2022-02-01
revenue HDI - Cohort 2022-07-01
94% HDI (train)
observed revenue (train)
94% HDI (test)
observed revenue (test)
train/pred split
2020-01
2020-07
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01
period

```
    x = pm.Gamma(
        name="revenue_estimated",
        alpha=n_active_users_estimated + eps,
        beta=lam,
        observed=revenue,
        dims="obs",
    )

    # --- Derived Quantities ---
    mean_revenue_per_user = pm.Deterministic(
        name="mean_revenue_per_user", var=(1 / lam), dims="obs"
    )
    pm.Deterministic(
        name="mean_revenue_per_active_user",
        var=p * mean_revenue_per_user,
        dims="obs"
    )
```



References

[1] Fader, P., Hardie, B., and Lee, K. "Counting Your Customers" the Easy Way: An Alternative to the Pareto/NBD Model. *Marketing Science 24* (05 2005), 275–284.

[2] Fader, P. S., and Hardie, B. G. How to project customer retention. *Journal of Interactive Marketing 21*, 1 (2007), 76–90.

[3] Fader, P. S., and Hardie, B. G. Fitting the sBG Model to Multi-Cohort Data. `<http://brucehardie.com/notes/017/`, 2017.

[4] Orduz, J. A Simple Cohort Retention Analysis in PyMC. `https://juanitorduz.github.io/retention/`, 12 2022.

[5] Orduz, J. Cohort Retention Analysis with BART. https://juanitorduz.github.io/retention_bart/, 01 2023.
[6] Orduz, J. Cohort Revenue & Retention Analysis: A Bayesian Approach. https://juanitorduz.github.io/revenue_retention/, 01 2023.
[7] Quiroga, M., Garay, P. G., Alonso, J. M., Loyola, J. M., and Martin, O. A. Bayesian additive regression trees for probabilistic programming, 2022.

Email address: juanitorduz@gmail.com
URL: https://juanitorduz.github.io/

Berlin, Germany