Product Analyst Assessment – Paywall A/B Test

Candidate: Samy Benmahrez

1. Context and Data

This analysis evaluates an A/B test of two paywall variants in a mobile app: *Standard* and *Plans*. The goal is to understand impact on (1) conversion from active users to paying users, (2) revenue net of taxes and platform commissions, and (3) retention via subscription renewals.

Two CSV datasets were provided: `user_behaviour.csv` (event-level telemetry with event_date, event_name, test_variant, paywall_type, product_identifier, and user identifiers) and `revenuecat.csv` (subscription-level data with product_identifier, price_in_usd, tax_percentage, commission_percentage, renewal_number, refund flags, and timestamps). All transformations and aggregations were done in a Python notebook (pandas) and exported to KPI summary CSVs, then used to build a Looker Studio dashboard.

2. Methodology and KPI Definitions

User identity: `revenue_cat_id` is the main user key across both datasets.
Variant assignment: each `revenue_cat_id` is assigned a single test_variant (Standard or Plans) based on events in `user_behaviour`.

2.1 Conversion Rate

Conversion rate (per variant):

$$\text{Conversion}_v = \frac{\#\{\text{users with at least one } \textit{payment\_success}\}}{\#\{\text{users with any event in variant } v\}}$$

Users are distinct by `revenue_cat_id`; all exposed users are the denominator and `payment_success` is the conversion signal.

2.2 Purchases by Paywall

Purchases by paywall type = number of distinct users with `payment_success`, grouped by (`test_variant`, `paywall_type`). This shows which paywall surfaces drive revenue per variant.

2.3 Revenue, ARPU, and ARPPU

For each non-refunded subscription row in RevenueCat:

$$\text{net\_revenue} = \text{price\_in\_usd} \times \big(1 - \text{tax\_percentage} - \text{commission\_percentage}\big)$$

Using this:

$$\text{Total net revenue}_v = \sum \text{net\_revenue}_i \quad \text{for subscriptions in variant } v$$
$$\text{ARPU}_v = \frac{\text{Total net revenue}_v}{\text{users}_v}$$
$$\text{ARPPU}_v = \frac{\text{Total net revenue}_v}{\text{paying users}_v}$$

Refunded subscriptions (`refunded_at` not null) are excluded.

2.4 Most Purchased Package

Product identifiers include suffixes _a/_b/_c. Normalise to a base SKU:

$$\text{product\_base} = \text{product\_identifier without trailing } a/b/c$$

Count distinct subscribers per (`test_variant`, `product_base`) to find the dominant package.

2.5 Retention via renewal_number

For each user and variant, compute max_renewal = max(renewal_number) across non-refunded subscriptions. Then per variant:

$$\text{Retention } k+ = \frac{\#\{\text{paying users with max\_renewal} \geq k\}}{\#\{\text{paying users}\}}, \quad k \in \{1, 2, 3\}$$

These approximate renewal-based retention across billing cycles. In production these definitions would be validated with Product and Finance stakeholders before finalising.

3. Results – Standard vs Plans

3.1 Top-line KPIs

Variant	Users	Paying users	Conversion	Net revenue ($)	ARPU ($)	ARPPU ($)
Plans	1,637	260	15.88%	3,339.67	2.04	12.84
Standard	1,544	405	26.23%	3,804.54	2.46	9.39

Interpretation. Standard converts significantly more users (26.2% vs 15.9%), generates more net revenue ($3.80k vs $3.34k), and higher ARPU (2.46 vs 2.04). Plans monetises each payer more (ARPPU 12.84 vs 9.39) but on fewer payers. Overall, Standard is the stronger default: more users convert, more renew, and total revenue is higher even though Plans extracts more per payer.

4. Paywalls and Packages

4.1 Purchases by Paywall Type

Using `kpi_purchases_by_paywall`, onboarding is the main driver of purchases in both variants; feature and laggard paywalls are secondary. Plans is lower across all paywall types, consistent with its lower conversion. **Implication:** the onboarding paywall is the highest-leverage surface (copy, pricing, layout, risk reduction).

4.2 Most Purchased Package

From `kpi_packages` (normalised `product_base`), the dominant package is `gala_1wt_1w_gold` in both variants (394 subscribers in Standard, 219 in Plans). Other packages (`gala_1w_creator`, `gala_1y_creator`, `gala_1y_gold`) have much smaller volume. **Implication:** revenue is concentrated on one core weekly Gold offer; optimising this SKU is the most leveraged path.

5. Retention

From `kpi_retention` (based on `max_renewal`):

- **Standard:** paying users 402; retention 1+ $\approx$ 22.6%; 2+ $\approx$ 11.7%; 3+ $\approx$ 6.0%.
- **Plans:** paying users 251; retention 1+ $\approx$ 20.3%; 2+ $\approx$ 7.6%; 3+ $\approx$ 2.4%.

Standard converts more and retains a higher share through successive renewals; Plans drops off faster after the first renewal, limiting LTV despite higher ARPPU.

6. Synthesis and Recommendations

- **Set Standard as the default** variant: it wins on conversion, total net revenue, ARPU, and renewal-based retention.
- **Use Plans selectively** for high-intent segments (heavy users, high-WTP geos/channels) to leverage its higher ARPPU.
- **Focus optimisation on onboarding** paywalls: value messaging, trial/guarantee, social proof, and showing value before asking.
- **Double down on** `gala_1wt_1w_gold`: experiment with price sensitivity, framing, limited-time offers, or bundles anchored on the weekly Gold plan.
- **Future analyses:** segment by country/acquisition channel; cohort-based retention by start month and variant; evaluate non-revenue impacts if available.

7. Looker Studio Dashboard

KPIs and breakdowns are exposed in an interactive dashboard (filters for variant and date; pages for overview, paywalls & packages, retention). Dashboard link: https://lookerstudio.google.com/reporting/4702f26e-4c79-4142-85d8-5dc097929d57.

8. Bonus – Churn flows leading to `rc_cancellation`

To understand how users exit, I analysed event sequences leading up to the first `rc_cancellation` per churned user.

Method (brief). Filtered users with `rc_cancellation`; sorted their events by `event_timestamp` and truncated at the first cancellation; collapsed consecutive duplicate events; kept the last 4 events (3 preceding + `rc_cancellation`); aggregated identical sequences into flows; reported the top 10 flows by user count.

Flow	Users
tool_opened → show_paywalls → skip_paywalls → rc_cancellation	27
image_generated → press_generate → image_generated → rc_cancellation	26
show_paywalls → skip_paywalls → avatar_creation → rc_cancellation	19
payment_initiated → payment_cancelled → skip_paywalls → rc_cancellation	17
skip_paywalls → show_paywalls → skip_paywalls → rc_cancellation	16
payment_success → ftue_landmarkt → tool_opened → rc_cancellation	14
skip_paywalls → avatar_creation → tool_opened → rc_cancellation	13
show_paywalls → payment_initiated → payment_cancelled → rc_cancellation	12
show_paywalls → skip_paywalls → show_paywalls → rc_cancellation	12
avatar_creation → show_paywalls → skip_paywalls → rc_cancellation	11

What this suggests.

- Many churners oscillate around paywalls (`show_paywalls`/`skip_paywalls`), hinting at paywall fatigue or weak value perception before exit.
- Payment friction appears: `payment_initiated` → `payment_cancelled` flows show intent blocked by cancellations—high-value paths to investigate for UX, pricing, or errors.

- Feature-then-churn flows (e.g., `image_generated`, `avatar_creation`) suggest users try AI/creative tools but churn soon after—either value is satisfied quickly or subscription value is unclear.

How I would use this as a PM.

- Reduce paywall fatigue: throttle repeats, vary messaging, show incremental value before re-showing.
- Fix payment friction: audit payment error logs/cancellation reasons, simplify steps, add recovery UX for retries.
- Link features to subscription value: in-feature prompts clarifying benefits, consider time-limited trials after engagement thresholds.