REPORT

BASIC EDA

Dimensions Exploration

To establish a foundational understanding of the dataset, we began by exploring key categorical dimensions namely, platform, network, and country from the install data; status from level completion events; package_type from revenue data; and network again from cost data. These dimensions represent essential non-numeric attributes that shape user behavior and app performance across channels and regions.

Our approach involved executing SELECT DISTINCT and ARRAY_AGG(DISTINCT ...) queries to enumerate unique values within each category. This allowed us to assess the variety and coverage of the data, identify potential segmentation variables, and lay the groundwork for downstream analyses such as behavioral trends by platform or monetization performance by acquisition channel. This step ensures data completeness and analytic flexibility in later stages.

Date Exploration

As part of our preliminary data validation, we conducted a comprehensive temporal analysis across all core tables in the project. This step focused on identifying the minimum and maximum event timestamps and calculating the total date range covered by each dataset. Using SQL functions such as MIN(), MAX(), and DATE_DIFF(), we determined the start and end dates, as well as the duration (in days) for each table. This approach allows us to validate the completeness of the data, identify any temporal gaps, and ensure chronological alignment across datasets. Establishing this time-based context is critical for accurate cohort, trend, and funnel analyses. For instance, confirming that install, session, and revenue data overlap is essential for evaluating user conversion paths and long-term value (LTV).

	table_name	min_date	max_date	date_range_days
0	q1_table_revenue	2021-04-30	2021-06-14	45
1	q1_table_install	2021-04-30	2021-06-01	32
2	q1_table_cost	2021-05-01	2021-05-31	30
3	q1_table_session	2021-04-30	2021-06-14	45
4	q1_table_level_end	2021-04-30	2021-06-14	45

Measures Exploration

To assess user engagement, monetization, and acquisition effectiveness, we performed a thorough exploration of key quantitative measures across the core datasets. This included install counts by country and platform, gameplay outcomes such as level wins and quits, session-based metrics like time spent and session frequency, revenue from purchases, and marketing costs broken down by network, country, and platform.

Our approach relied on SQL aggregation functions including COUNT(), SUM(), AVG(), MIN(), MAX(), and conditional aggregations via COUNTIF() to isolate specific behaviors. We also applied CAST() functions where type conversion was needed, particularly for monetary values. This measure-level analysis enables us to establish performance baselines, calculate advanced KPIs such as ARPU or ROAS, and surface areas of inefficiency such as disproportionate spend with low conversion. These metrics are critical to quantifying the scale and impact of both product and marketing efforts.

Magnitude Exploration

To evaluate the relative scale and business impact of different segments, we conducted a magnitude analysis across key dimensions. This included examining installs by platform, network, and country; sessions by platform; revenue by platform and package type; and advertising cost across network, country, and platform. By aggregating these performance indicators, we aimed to identify dominant contributors, high-impact user groups, and potential imbalances in performance.

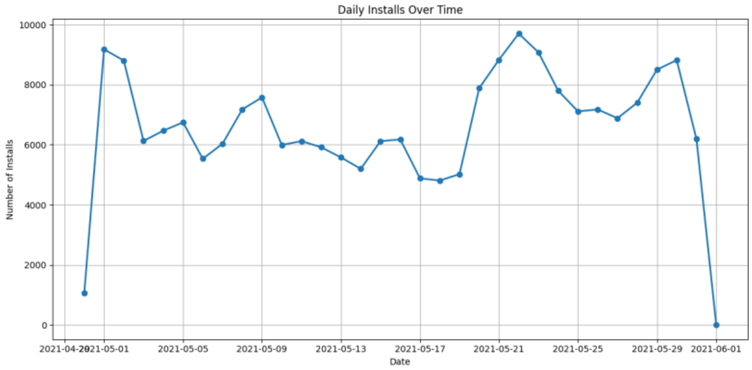
Our method relied on SQL aggregation functions such as SUM(), COUNT(), AVG(), MIN(), and MAX(), grouped by relevant categorical fields. This analysis highlights where the majority of user activity, revenue, or costs are concentrated supporting data-driven decisions for budget allocation, platform prioritization, and strategic optimization. It also helps reveal inefficiencies, such as channels with high spend but low returns, enabling focused corrective actions.

ADVANCED EDA

Change-Over-Time Analysis

Change-over-time analysis focuses on monitoring how key metrics evolve on a daily basis, offering insights into user behavior, engagement, and monetization dynamics. This time-series approach allows us to track the progression of core indicators such as installs, level completions, sessions, revenue, and marketing costs. By grouping each metric by date, we generate a chronological view that supports trend identification, anomaly detection, and the evaluation of campaign or product update impacts.

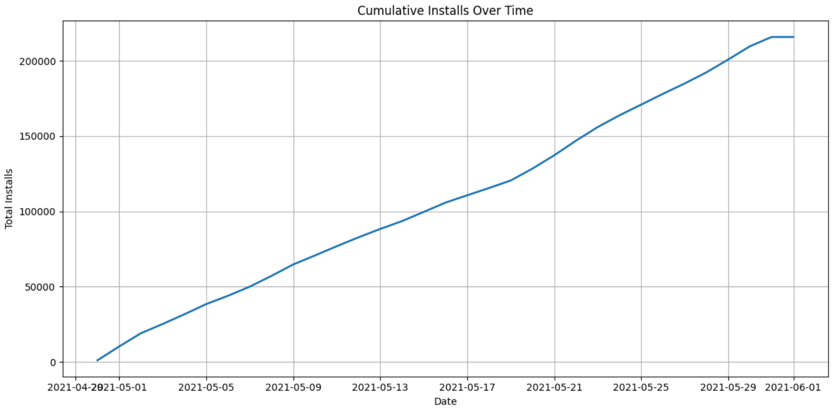
Our method utilizes SQL queries to calculate daily aggregates: query_cot_1 counts unique daily installs; query_cot_2 captures daily level completions and exits; query_cot_3 records daily user sessions; query_cot_4 summarizes in-app purchase revenue by day; and query_cot_5 tallies daily marketing expenditures. These time-based measurements are essential for understanding product momentum, seasonal shifts, and overall business health enabling informed, timely decision-making.



Cumulative Analysis

Cumulative analysis tracks the running total of key metrics over time, offering a clear picture of long-term performance and growth. By summing daily values sequentially, this method highlights the pace and sustainability of user acquisition, engagement, monetization, and marketing investment. Unlike daily snapshots, cumulative views provide full-period insights that support more strategic, big-picture decision-making.

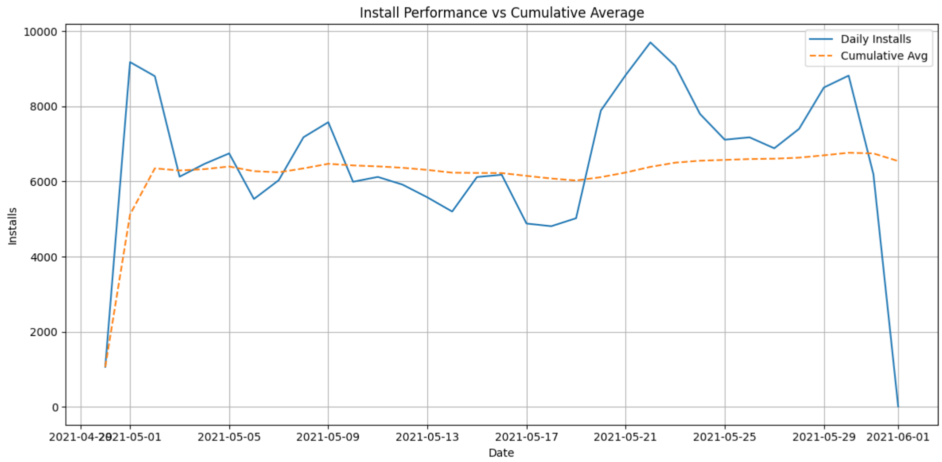
To perform this analysis, we utilized SQL window functions such as SUM(...) OVER (ORDER BY date), which compute the cumulative value of each metric across time. Specifically, query_ca_1 captures cumulative installs, query_ca_2 tracks total level completions, query_ca_3 measures cumulative sessions, query_ca_4 reflects total in-app revenue, and query_ca_5 sums up marketing spend to date. These cumulative metrics reveal growth trajectories, help identify plateaus, and serve as a foundation for evaluating overall product and business performance.



Performance Analysis

Performance analysis assesses daily metrics in relation to their cumulative averages, enabling the identification of deviations from historical trends. This approach helps surface whether a specific day's performance such as installs, sessions, or revenue is outperforming or underperforming relative to past behavior, providing a dynamic benchmark for operational decision-making.

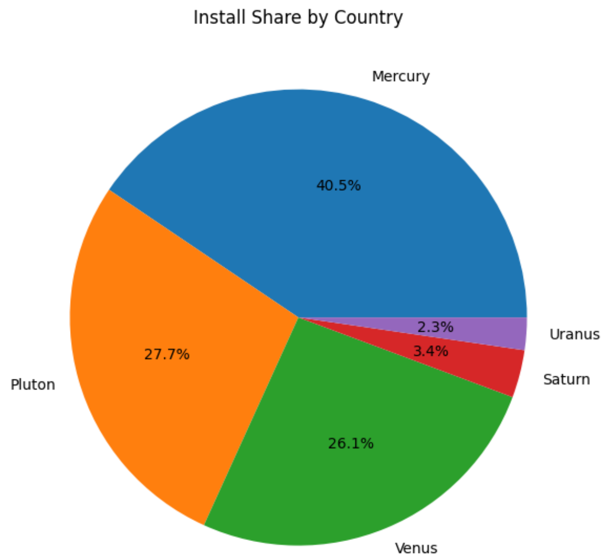
Using SQL window functions like `AVG(...) OVER (...)`, each query computes the daily metric, its cumulative average up to that point, and the gap between the two. For example, `query_pa_1` measures install trends against average acquisition, `query_pa_2` captures gameplay fluctuations via level completions, `query_pa_3` assesses engagement consistency through session activity, `query_pa_4` flags changes in monetization patterns, and `query_pa_5` compares daily ad spend to historical pacing. This methodology enables early detection of anomalies, supports agile marketing or product adjustments, and ensures performance is continuously evaluated against evolving expectations.



Part-to-Whole Analysis

Part-to-whole analysis evaluates how individual segments contribute to the overall dataset, providing clarity on the relative importance of categories such as countries, platforms, networks, and package types. This approach enables teams to identify leading contributors, assess market concentration, and make informed strategic decisions about resource allocation and focus areas.

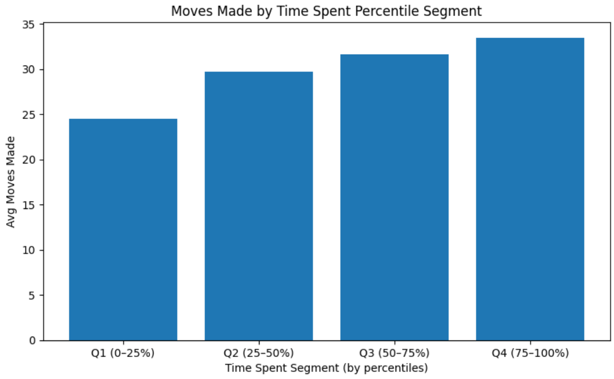
Using standard SQL aggregation functions combined with percentage calculations like ROUND(... * 100.0 / SUM(...) OVER (), 2), we quantify each category's share of the total. Specifically, query_ptw_1 identifies top-performing countries in user acquisition, query_ptw_2 outlines the distribution of level outcomes (win, fail, quit), query_ptw_3 highlights platform usage patterns (with typo correction for accuracy), query_ptw_4 surfaces high-revenue package types, and query_ptw_5 breaks down marketing cost by network. This method reveals where value is concentrated and guides prioritization across markets, platforms, and monetization strategies.



Data Segmentation Analysis

Data segmentation analysis involves dividing datasets into value-based subgroups typically using percentile ranges such as quartiles to uncover behavioral or performance differences across segments. This method helps move beyond high-level aggregates to reveal nuanced patterns and inform more tailored strategies.

Using percentile-based logic, each record is classified into quartiles (Q1–Q4), and key metrics are analyzed within those groups. For example, query_ds_1 segments users by time spent in a level and evaluates corresponding move counts, while query_ds_2 explores the relationship between session duration and coin balance. query_ds_3 groups purchases by revenue tiers to understand spending behaviors, and query_ds_4 analyzes marketing cost distribution across spend levels. This segmentation framework enables meaningful comparisons between high- and low-performing cohorts, supporting precision in marketing, monetization, and product optimization efforts.



TASK 1

DAU

Daily Active Users (DAU) measures the number of unique users who engage with the application each day. It serves as a core indicator of user engagement, product stickiness, and growth momentum. Calculated by counting distinct user IDs interacting with the app over a 24-hour period, DAU offers valuable insight into behavioral trends and platform health. When segmented by attributes such as platform, country, or acquisition network, it enables deeper evaluation of audience dynamics and campaign impact.

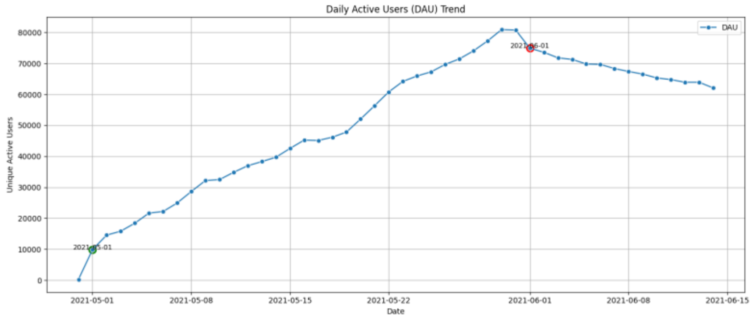
Trend Summary: DAU showed steady growth throughout May, peaking at approximately 80,000 users at the end of the month. A sharp decline was observed on June 1st, followed by a gradual downward trend. Anomaly detection indicated significant short-term spikes and dips, likely linked to marketing campaigns or system-level changes.

By Platform: iOS consistently outperformed Android in terms of DAU. Both platforms peaked in early June; however, Android experienced a steeper decline post-peak compared to iOS, suggesting potential retention or engagement issues.

By Country: Mercury contributed the highest DAU and peaked at the end of May. Pluton saw rapid growth but declined sharply post-peak, while Venus maintained stable mid-level activity with a gradual drop in June. Saturn and Uranus had minimal engagement throughout the period.

By Network: Buzz was the top-performing ad network, reaching nearly 49,000 daily users before declining gradually. Organic users showed strong and sustained engagement, ranking second overall. Woody maintained consistent but moderate activity, whereas Jessie and Sid had negligible impact. All networks followed a similar pattern of peaking in late May to early June.

This analysis enables targeted strategy refinement across marketing, platform optimization, and user retention initiatives by highlighting where active engagement is strongest and where drop-offs occur.



ARPU

ARPU measures the average revenue generated per user over a defined period, serving as a key metric for evaluating user value and monetization efficiency. In this analysis, we focus on 7-day ARPU, which assesses revenue earned within the first 7 days post-install. This variation is particularly useful for understanding early monetization patterns and guiding acquisition and retention strategies.

Calculation:

7-Day ARPU is computed as:

ARPU = Total Revenue within 7 Days / Total Number of Users

For this dataset, 215,082 users generated approximately $132,280 in revenue within their first week, resulting in a 7-day ARPU of $0.615. This reflects moderate monetization during the early lifecycle phase.

By Country: Mercury and Venus led with 7-day ARPU values above $0.80, indicating high early revenue potential in these regions. In contrast, Uranus, Pluton, and Saturn recorded significantly lower ARPU, suggesting limited short-term monetization. These insights support concentrating marketing spend and feature rollouts in high-performing regions like Mercury and Venus for better ROI.

By Platform: iOS users exhibited nearly three times the ARPU of Android users, signaling stronger purchasing behavior and higher revenue yield on Apple devices. This platform-level disparity highlights an opportunity for platform-specific strategies, especially to improve early-stage monetization on Android.

By Network: The Sid network delivered the highest ARPU—nearly triple that of other networks—demonstrating superior user quality or campaign effectiveness. Buzz, Organic, Jessie, and Woody performed at similar, lower levels. Prioritizing Sid for user acquisition could significantly enhance early revenue outcomes.

This ARPU analysis supports data-driven decision-making in marketing spend, product targeting, and monetization design by revealing which segments generate the most value within the critical early user window.

ARPDAU

ARPDAU quantifies daily revenue efficiency by measuring the average revenue generated per active user each day. It integrates engagement and monetization into a single metric, offering a powerful lens to evaluate the financial performance of the app at the user level. ARPDAU is calculated by dividing daily revenue by DAU:

ARPDAU = Total Revenue on a Day / DAU on the Same Day

This metric is essential for understanding monetization effectiveness, especially when segmented by platform, country, or acquisition network. It supports decision-making on pricing, in-app purchase design, advertising intensity, and user acquisition strategies.

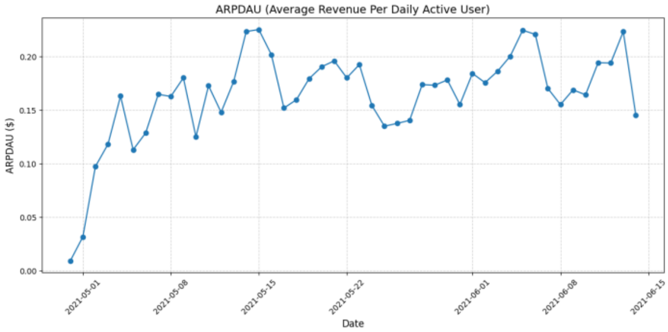
Trend Summary: ARPDAU exhibited moderate variability throughout the analysis period, with peaks in mid-May and early June. Sustained values above $0.15 suggest healthy monetization, while intermittent dips may reflect declines in user spending or engagement.

By Platform: iOS consistently outperformed Android in ARPDAU, indicating that iOS users generate more revenue per active session. This gap suggests that monetization strategies should be tailored to boost revenue yield on Android, where performance lags.

By Country: Venus and Mercury demonstrated the highest ARPDAU values across the period, signaling effective monetization in those markets. In contrast, Saturn, Uranus, and Pluton recorded substantially lower ARPDAU. A sharp, isolated spike in Uranus suggests either a one-time high spender or a data anomaly. These disparities point to the need for localized monetization strategies to maximize per-user revenue.

By Network: The Sid network achieved the highest ARPDAU with multiple revenue spikes, suggesting strong short-term monetization but with volatility—potentially linked to aggressive campaign tactics or short-lived user cohorts. Jessie had the weakest performance, with consistently low ARPDAU. Buzz, Organic, and Woody maintained stable, mid-range values, indicating dependable revenue per user without major fluctuations.

This analysis enables teams to optimize monetization strategies based on platform, geography, and acquisition source, ensuring revenue opportunities are maximized across user segments.



ROAS

ROAS measures the efficiency of advertising investments by evaluating how much revenue is generated for each dollar spent. It is a core metric for marketing performance, guiding decisions on campaign optimization, budget allocation, and strategic growth. The formula used is:

ROAS = Total Revenue / Total Advertising Cost

A ROAS above 1.0 (or 100%) indicates profitability, while values below this threshold suggest underperformance.

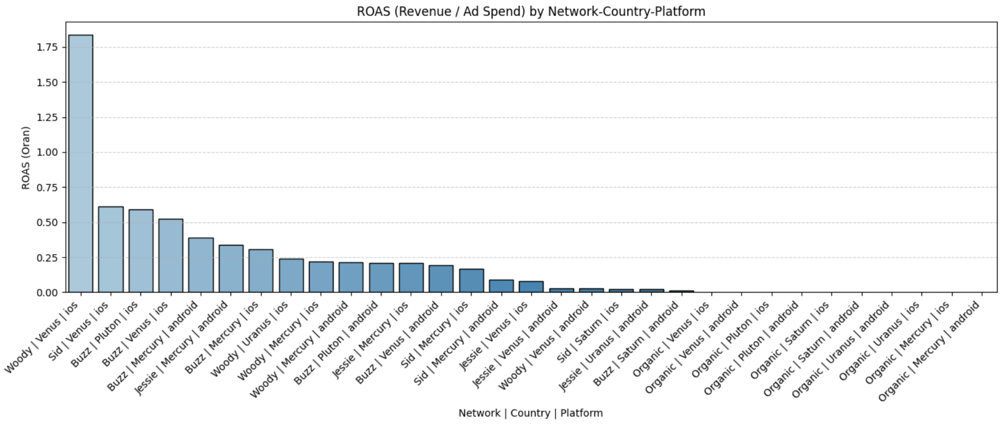
By Channel (Network–Country–Platform): The highest ROAS was recorded for Woody | Venus | iOS at 1.83, signaling an exceptionally strong return on ad spend. Other high performers include Sid | Venus | iOS and Buzz | Pluton | iOS, all on the iOS platform—highlighting its efficiency in converting paid users into revenue. Overall, iOS channels consistently outperformed Android in ROAS across all segments.

By Platform: iOS achieved a ROAS of 0.50, significantly higher than Android's 0.32. This indicates that iOS users yield better returns per dollar spent, suggesting that shifting more marketing budget to iOS could improve cost-efficiency and overall revenue impact.

By Country: Uranus and Saturn delivered the strongest ROAS—around 3.0—indicating highly effective user acquisition and monetization in these regions. In contrast, countries like Pluton, Venus, and Mercury fell below 0.6, signaling lower revenue return per ad dollar. This geographic variance supports concentrating spend in high-ROAS markets such as Uranus and Saturn.

By Network: Sid and Buzz networks emerged as top performers with ROAS values above 0.35, showcasing strong cost-efficiency. Woody and Jessie trailed behind. Sid, in particular, appears to be the most effective paid acquisition channel, consistently delivering better returns across multiple dimensions.

This analysis provides clear guidance for reallocating marketing budgets toward high-performing segments and refining acquisition strategies for maximum return on investment.



Retention

To assess user engagement and long-term value, we conducted a retention analysis focusing on Day 1, Day 3, and Day 7 checkpoints. Retention is calculated as the percentage of users who return on a given day post-install, with segmentation by platform, country, and acquisition network to identify patterns in user loyalty. Our method tracked returning users over time and evaluated cohort behavior by comparing install dates with subsequent activity.

The overall retention performance is stable across the observed period. Day 1 retention ranges from 52–58%, Day 3 from 43–46%, and Day 7 from 38–41%, indicating consistent user stickiness beyond initial use. iOS users exhibit higher Day 7 retention (40–43%) compared to Android (34–39%), suggesting a more loyal user base on Apple devices. Country-level analysis highlights Pluton and Mercury as top-performing regions with 42–45% retention, while Venus and Uranus underperform with rates around 30–38%. Saturn stabilizes near 40% after early variation.

From a network perspective, Sid leads with Day 7 retention frequently exceeding 45–50%, reflecting strong acquisition quality. Buzz and Organic sources perform reliably at 38–42%, while Jessie shows erratic retention patterns, including 0% on multiple dates—indicating potential campaign or data issues. Woody trends downward from ~35% to ~26%, warranting optimization. Overall, retention analysis reveals clear opportunities to reinforce high-performing segments like iOS and Sid, while underperforming channels such as Jessie require immediate attention.

Churn

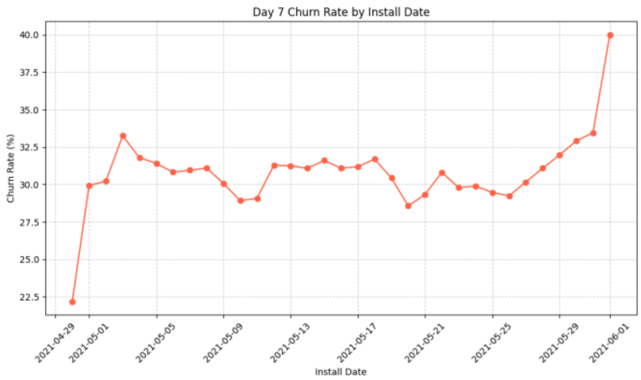
Churn represents the proportion of users who stop engaging with the app within a defined time window. For this analysis, we focus on Day 7 churn, which captures users who do not return between Day 1 and Day 7 after installation. This metric serves as the inverse of retention and is calculated as:

Churn Rate = (Users Who Did Not Return by Day 7) / (Total Users) × 100

This approach complements retention analysis and provides critical insights into user disengagement. It helps pinpoint issues in onboarding, user experience, or targeting that may be leading to early drop-offs.

Throughout the observed period, the Day 7 churn rate remains relatively stable between 30–33%, indicating consistent user drop-off behavior. However, a notable spike to 40% on June 1 suggests a potential issue—possibly linked to low-quality user acquisition, a technical disruption, or the launch of an ineffective campaign. Conversely, the lowest churn rate, 22% on April 29, may reflect a high-retention user cohort or a particularly effective acquisition source on that day.

These findings emphasize the need to investigate the campaigns or channels active on June 1 to identify the cause of elevated churn. Likewise, replicating conditions from low-churn days may help improve early user retention and reduce future drop-off rates.



LTV

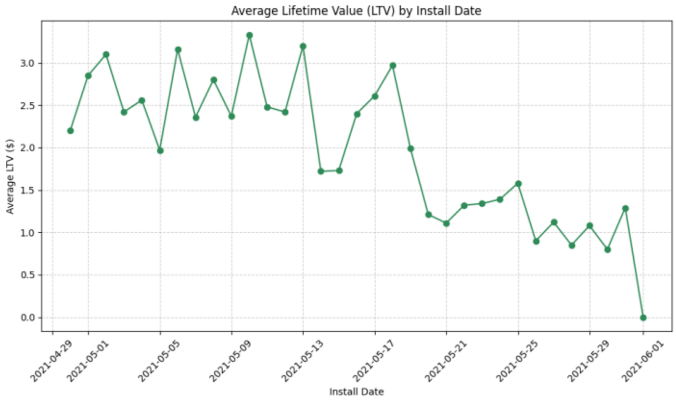
Lifetime Value (LTV) measures the total revenue a user is expected to generate over their lifetime within the app. It is a core metric for evaluating monetization effectiveness, marketing efficiency, and long-term business sustainability. LTV is calculated as:

LTV = Total Revenue from Users / Number of Users

To compute LTV by cohort, we joined install and revenue data, aggregating revenue per user and grouping by install date to derive average LTV values over time. This method provides visibility into the financial value of users acquired on specific days and highlights shifts in user quality or monetization effectiveness.

Key Insights: From April 29 to May 20, average LTV values remained strong—ranging between 2.0 and 3.3, indicating high monetization efficiency. However, starting around May 21, LTV dropped sharply, falling below 1.5 and eventually reaching 0.0 on June 1. This steep decline signals a potential issue, such as an influx of low-quality users, a failed monetization experiment, or a change in product or campaign strategy.

Next Steps: Further investigation is needed to identify what changed after mid-May. Reviewing acquisition sources, campaign targeting, and in-app purchase configurations could uncover the cause behind the revenue collapse. Addressing these issues promptly is critical to restoring LTV performance and sustaining profitable growth.



TASK 2

Monetization A/B Test

To evaluate the effectiveness of alternative monetization strategies, a controlled A/B test was conducted. Users were randomly assigned to two groups Group A (control) and Group B (test) to compare revenue performance under different monetization conditions. Key metrics such as conversion rate, ARPU, revenue per paying user, and time to first purchase were tracked and analyzed across both groups. The goal was to identify whether the test variant (Group B) produced a statistically and practically significant improvement in monetization outcomes.

The results clearly indicate that Group B outperformed Group A on all monetization metrics. Conversion rate increased by over 20%, total revenue rose by approximately 27%, and both ARPU and revenue per paying user were meaningfully higher. Additionally, Group B users made their first purchase sooner, suggesting stronger early engagement. These findings point to Group B as the superior variant, with improved monetization efficiency and user responsiveness. Further statistical validation is recommended before a full-scale rollout, but early evidence supports prioritizing Group B for broader deployment.

	group_id	total_users	paying_users	conversion_rate	total_revenue	arpu	revenue_per_paying_user	avg_days_to_first_purchase
0	A	37188	2782	7.48	439673.0	11.82	158.04	6.28
1	B	36262	3261	8.99	552916.0	15.25	169.55	5.54

Engagement A/B Test

An Engagement A/B Test was conducted to assess how different product experiences impact user interaction and behavior. Users were randomly segmented into Groups A and B, each exposed to a different version of the app. The focus of this test was on engagement not revenue—using key metrics such as session count, total and average time spent, session duration, and average progression (levels reached) to evaluate performance.

Results show that Group A demonstrated significantly higher engagement. Total sessions increased by 33%, time spent per user rose by 31%, and average level progression nearly doubled compared to Group B. Session length remained similar across both groups, indicating that the core experience was stable, but Group A users returned more frequently and progressed further. Although Group B yielded stronger monetization in previous tests, these findings suggest that changes in that variant may have unintentionally reduced engagement. Moving forward, it is essential to consider whether revenue gains in Group B come at the cost of long-term user activity and satisfaction.

	group_id	total_users	total_sessions	avg_sessions_per_user	total_time_spent	avg_time_spent_per_user	avg_time_per_session	avg_level_reached
0	A	37188	128173351	3446.63	3.706398e+09	99666.51	28.92	477.62
1	B	36262	95970948	2646.60	2.758156e+09	76061.89	28.74	271.96

TASK 3

EDA

The EDA process begins by inspecting data types within df_ml_1 to ensure compatibility with downstream machine learning tasks. Irrelevant identifiers like user_id are removed to avoid introducing noise into the model. Missing values are checked across columns, and categorical variables with object dtype are encoded using label encoding. Age values are further segmented into defined age groups to enhance interpretability and predictive value. The script also examines the number of unique values in key columns to assess cardinality, aiding in feature engineering decisions.

For outlier detection, boxplots are generated for all features except the target (d30_revenue), providing a visual overview of value distributions and highlighting potential anomalies. Descriptive statistics are displayed in a transposed format for easier analysis, followed by normalization of features (excluding the target) via Min-Max scaling to standardize the input space for modeling.

In the distribution and correlation phase, histograms, KDE plots, and count plots are used to visualize feature distributions based on their data type. A correlation matrix and heatmap are created to identify linear relationships, particularly multicollinearity, and feature-target correlations. The d30_revenue target is converted into a binary variable to support classification modeling, and class balance is checked via a pie chart.

Data visualization tools such as scatter plots illustrate potential relationships between numerical features and the target. Finally, the evaluation module includes functions to assess classifier performance using accuracy, confusion matrix heatmaps, classification reports, and ROC curves with AUC annotations allowing for effective comparison of model performance across multiple algorithms.

Models

The modeling phase begins by splitting the dataset into training and test sets using an 80/20 ratio to ensure robust evaluation and prevent overfitting. Multiple classification algorithms are applied to predict the binary version of the d30_revenue target, each offering a different perspective on performance and learning complexity.

The pipeline includes both baseline and advanced models. Logistic Regression serves as a foundational benchmark for linear separability. K-Nearest Neighbors (KNN), Naive Bayes, and Decision Tree Classifiers provide interpretable yet varied learning strategies. Random Forest and Gradient Boosting offer ensemble-based improvements, often delivering stronger generalization by reducing variance or bias.

To capture non-linear and high-dimensional patterns, neural network models are implemented. A standard Neural Network is evaluated first, followed by a Weighted Neural Network, which incorporates class weighting to handle any class imbalance in the binary target.

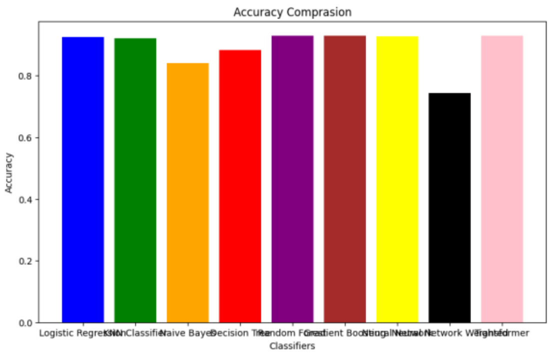
Finally, a Transformer-based model is tested, leveraging attention mechanisms for potentially deeper pattern recognition especially valuable in feature-rich, complex datasets.

Each model's performance is assessed consistently using predefined evaluation functions, enabling clear comparison across accuracy, ROC-AUC, and classification metrics.

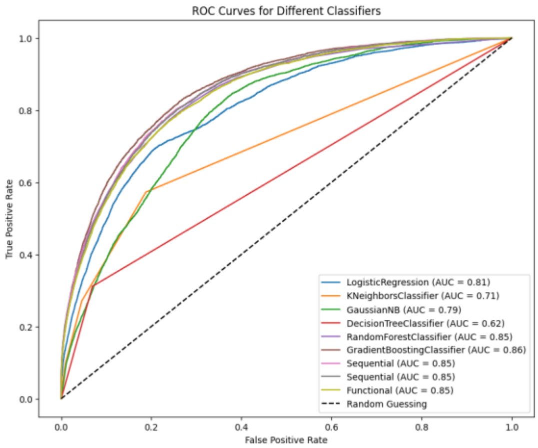
Evaluation

To compare model performance, we evaluated all classifiers using two key metrics: accuracy and AUC-ROC (Area Under the Receiver Operating Characteristic Curve). Accuracy reflects the overall correctness of the predictions, while AUC provides a more robust measure of a model's ability to distinguish between classes, especially in imbalanced datasets.

Accuracy Comparison: The accuracy chart reveals strong performance across most models, with Random Forest, Gradient Boosting, and the Transformer-based model achieving the highest scores (~94%). Logistic Regression, KNN, and Neural Network also performed well, hovering just below that threshold. In contrast, Naive Bayes and Weighted Neural Network lagged behind, indicating either limitations in learning capacity or misalignment with the data distribution.



ROC Curve Analysis: The ROC curves further validate model performance. Gradient Boosting achieved the highest AUC at 0.86, closely followed by Random Forest, Transformer, and Neural Network models (all at 0.85). Logistic Regression also performed reliably with an AUC of 0.81, while Naive Bayes achieved 0.79. KNN and Decision Tree underperformed in comparison, with AUC scores of 0.71 and 0.62, respectively, indicating weaker class separation capabilities.



Comments: Ensemble models and neural architectures consistently outperformed simpler models in both accuracy and AUC, highlighting their ability to capture complex patterns. While Logistic Regression remains a strong baseline, models like Gradient Boosting and Transformer offer superior predictive power. Based on these results, deployment should prioritize these top-performing models, with further tuning or ensembling considered for marginal gains.