TravelTide

Loyalty Program

Executive Summary

Introduction

This executive summary outlines the project to develop and implement a new loyalty program for TravelTide. The primary goal of this project is to enhance customer engagement and retention by assigning tailored perks to different customer segments based on their behavior, value, and loyalty. The new program is designed to reward customers more effectively, drive increased spending, and strengthen brand loyalty.

Objective

□ To identify and categorize customers into meaningful segments.

□ To assign relevant and attractive perks that motivate each customer segment.

□ To increase customer retention, average spend per customer, and overall satisfaction.

□ To create a loyalty program that differentiates Traveltide in the marketplace.

Project Overview

To achieve these objectives, we segmented our customers into six distinct groups: Diamond, Platinum, Gold, Silver, Bronze, and Regular Customers. The segmentation is based on various criteria, including number of bookings, transaction value, and engagement level with the brand. Each group will receive a unique set of perks aligned with their specific preferences and value to the company.

Methodology

Our approach involved analyzing customer data to identify behavioral patterns and preferences. We used rules based approach to define the six customer segments. Based on these segments, we tailored specific perks and benefits that would appeal to each group, while also ensuring they align with the business goals and budget.

Key Findings

□ Diamond Customers : The highest-value segment represent 8% of the total number of users in the cohort.

□ Platinum Customers : represent 9% of the users

□ Gold Customers : represent 14% of the users

□ Silver Customers : represent 21% of the users

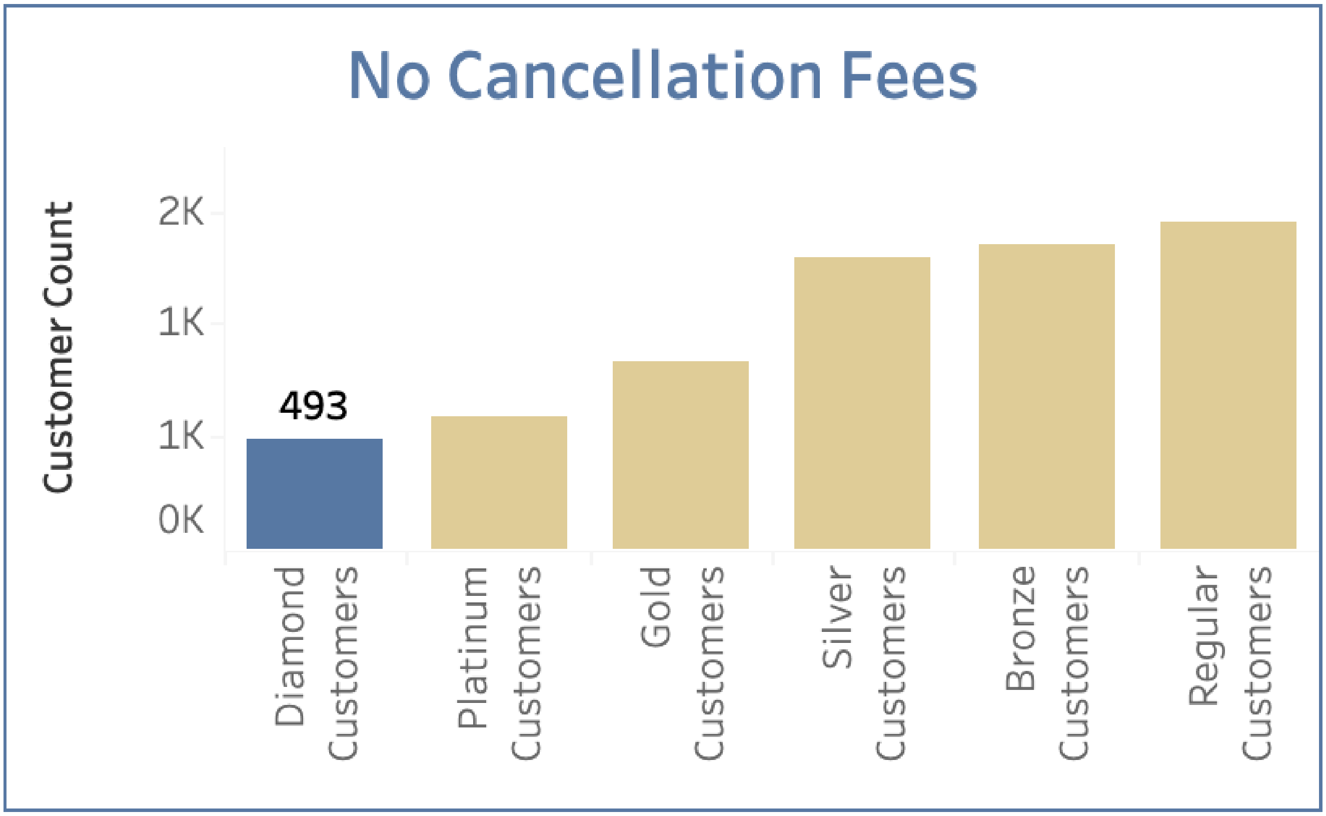
□ Bronze Customers : represent 22% of the users

□ Regular Customers : represent 25% of the users

Proposed Perks

Based on the customer segments, the following perks have been proposed:

□ Diamond Customers : No cancellation fees



Platinum Customers : 10% flight discount

alt text

Gold Customers: Free hotel night with a flight purchase

alt text

Silver Customers: 10% hotel discount

alt text

Bronze Customers: Free hotel meal

alt text

Regular Customers: Free Checked Bag

alt text

Recommendations

To ensure the success of the new loyalty program, we recommend:

▯ Implementing a phased rollout of the program to test and refine the perks.

▯ A/B testing and Continuously monitoring customer response

▯ Adjusting perks based on feedback and data analysis.

▯ Promoting the new program through targeted marketing campaigns to
maximize awareness and participation.

Conclusion

This project provides a strategic framework for enhancing customer loyalty and engagement through a differentiated rewards system. By aligning perks with customer value and preferences, we expect to see an increase in customer retention rates, higher number of bookings, higher average spending, and a stronger connection between customers and our brand. The new loyalty program represents a significant opportunity for TravelTide to drive growth and long-term customer satisfaction.

Annexe I : SQL Code

```
-- This CTE returns the session details after 4th Jan 2023 as per
Elena's cohort

WITH sessions_2023 AS (
    SELECT s.*
    FROM sessions s
    WHERE s.session_start > '2023-01-04'
),

-- This CTE returns the IDs of all users with more than 7 sessions as
per Elena's cohort

filtered_users AS (
    SELECT user_id,
           COUNT(*)
    FROM sessions_2023 s
    GROUP BY user_id
    HAVING COUNT(*) > 7
),

-- This CTE combines the data from different tables using joins

session_base AS (
        SELECT
        s.session_id,
        s.user_id,
        s.trip_id,
        s.session_start,
        s.session_end,
        s.page_clicks,
        s.flight_discount,
        s.flight_discount_amount,
        s.hotel_discount,
        s.hotel_discount_amount,
        s.flight_booked,
        s.hotel_booked,
        s.cancellation,
        u.birthdate,
        u.gender,
        u.married,
        u.has_children,
        u.home_country,
        u.home_city,
        u.home_airport,
        u.home_airport_lat,
        u.home_airport_lon,
        u.sign_up_date,
        f.origin_airport,
        f.destination,
        f.destination_airport,
        f.seats,
        f.return_flight_booked,
        f.departure_time,
        f.return_time,
        f.checked_bags,
        f.trip_airline,
        f.destination_airport_lat,
        f.destination_airport_lon,
        f.base_fare_usd,
        h.hotel_name,
    CASE
        WHEN h.nights < 0 THEN 1
        ELSE h.nights
    END AS nights,
        h.rooms,
        h.check_in_time,
        h.check_out_time,
        h.hotel_per_room_usd AS hotel_price_per_room_per_night_usd
    FROM sessions_2023 s
    LEFT JOIN users u
    ON s.user_id = u.user_id
    LEFT JOIN flights f
    ON s.trip_id = f.trip_id
    LEFT JOIN hotels h
    ON s.trip_id = h.trip_id
    WHERE s.user_id IN (SELECT user_id FROM filtered_users)
 ),

--This CTE returns the trip Ids of cancelled trips

cancelled_trips AS (
    SELECT DISTINCT trip_id
    FROM session_base
    WHERE cancellation = TRUE
    ),

--This CTE returns the trip IDS of the trips that are not in the
previous CTE which means that
--it returns only Trip Ids of the not cancelled trips

not_cancelled_trips AS (
    SELECT *
    FROM session_base
    WHERE trip_id is not null
    AND trip_id NOT IN (SELECT trip_id FROM cancelled_trips)
    ),

--This CTE returns aggregated metrics at the session level

user_base_session AS (
    SELECT user_id,
        SUM(page_clicks) AS num_clicks,
        COUNT(DISTINCT session_id) AS num_sessions,
        AVG(session_end - session_start) AS avg_session_duration
    FROM session_base
    GROUP BY user_id
),

--This CTE returns aggregated metrics at the user level

    user_base_trip AS (
    SELECT user_id,
        COUNT(DISTINCT trip_id) AS num_trips,
            SUM(seats) AS num_seats,
            CEILING(AVG(seats)) AS avg_seats,
            SUM(rooms) AS num_rooms,
            SUM(nights) AS num_nights,
        SUM(CASE
            WHEN (flight_booked = TRUE) AND (return_flight_booked =
TRUE) THEN 2
            WHEN flight_booked = TRUE THEN 1 ELSE 0
            END) AS num_flights,
            SUM(CASE
            WHEN (flight_booked = TRUE) AND (hotel_booked = TRUE) THEN 2
            WHEN flight_booked = TRUE THEN 1
            WHEN hotel_booked = TRUE THEN 1
            END) AS num_bookings,
            SUM(CASE
            WHEN hotel_booked = 'true' THEN 1 ELSE 0
            end) AS num_hotels,
            SUM(CASE
            WHEN flight_discount = 'true' AND flight_booked = 'true'
THEN 1 ELSE 0
```

```
            end) AS flt_disc,
            SUM(CASE
            WHEN hotel_discount = 'true' AND hotel_booked = 'true' THEN
1 ELSE 0
            end) as hotel_disc,
            COALESCE(
            SUM(
        hotel_price_per_room_per_night_usd * nights * rooms *
        (1 - COALESCE(hotel_discount_amount, 0))
            ), 0) AS money_spent_hotel,
            COALESCE(
            SUM(
        base_fare_usd *
        (1 - COALESCE(flight_discount_amount, 0))),0) AS
money_spent_flight,
            AVG(EXTRACT(DAY FROM departure_time - session_end)) AS
travel_lead_time,
            AVG(EXTRACT(DAY FROM return_time - departure_time)) AS
trip_length,
            AVG(haversine_distance(home_airport_lat, home_airport_lon,
destination_airport_lat, destination_airport_lon)) AS
avg_distance_travelled,
                SUM(checked_bags) as total_checked_bags
    FROM not_cancelled_trips
    GROUP BY user_id
 ),


--This CTE is used to calculate the number of cancellations per user

temp1 AS (
  SELECT sessions.user_id,
            SUM(CASE WHEN sessions.cancellation = 'true' THEN 1 ELSE 0
END) AS num_cancellations FROM sessions
    RIGHT JOIN filtered_users ON sessions.user_id =
filtered_users.user_id
  GROUP BY sessions.user_id
    ORDER BY sessions.user_id
),

--This CTE is used to bring together the session metrics and the user
metrics

metrics AS (
  SELECT b.*,
    EXTRACT(YEAR FROM AGE(u.birthdate)) AS age,
    u.gender,
    u.married,
    u.has_children,
    u.home_country,
    u.home_city,
    u.home_airport,
    u.home_airport_lat,
    u.home_airport_lon,
    t.num_trips,
    t.num_flights,
    t.num_hotels,
    t.num_nights,
    t.num_bookings,
    temp1.num_cancellations,
    (temp1.num_cancellations * 100 / t.num_bookings) as
cancellation_rate_perc,
        t.money_spent_hotel,
    t.money_spent_flight,
    t.flt_disc,
    t.hotel_disc,
    t.travel_lead_time,
    t.trip_length,
    t.trip_length/t.num_trips as avg_trip_length,
    t.num_seats,
    t.avg_seats,
    t.num_rooms as num_rooms,
    (t.money_spent_hotel + t.money_spent_flight)/t.num_trips AS
avg_spend,
    t.avg_distance_travelled,
    t.total_checked_bags

    FROM user_base_session b
    LEFT JOIN users u ON b.user_id = u.user_id
    LEFT JOIN user_base_trip t ON b.user_id = t.user_id
    LEFT JOIN temp1 ON b.user_id = temp1.user_id
),

--This CTE is used to create customer segments based on different
conditions of the calculated metrics

perks AS (
    SELECT *,
    CASE WHEN num_flights >=8 AND money_spent_flight>=1800 AND
cancellation_rate_perc = 0 THEN 'Diamond Customers'
             WHEN num_seats >=3 AND num_flights > 4 AND
avg_distance_travelled > 2000 THEN 'Gold Customers'
             WHEN (trip_length <=2 AND num_flights > 5 ) OR ( num_seats
>=4 AND num_flights > 4 AND avg_distance_travelled > 1200) THEN
'Platinum Customers'
             WHEN num_nights > 8 THEN 'Silver Customers'
             WHEN money_spent_hotel > 450 THEN 'Bronze Customers'
             ELSE 'Regular Customers'
    END as customer_Segment
    FROM metrics
)

-- This query is used to assign perks to the different customer segments

    SELECT customer_segment,COUNT(*) AS customer_count,
    CASE WHEN customer_segment = 'Diamond Customers' THEN 'No
Cancellation Fees'
             WHEN customer_segment = 'Platinum Customers' THEN '10%
Discount On Flight Bookings'
             WHEN customer_segment = 'Gold Customers' THEN 'One Night
Hotel Stay Free With Flight'
             WHEN customer_segment = 'Silver Customers' THEN '10%
Discount On Hotel Bookings'
             WHEN customer_segment = 'Bronze Customers' THEN 'One Free
Hotel Meal'
             WHEN customer_segment = 'Regular Customers' THEN 'One Free
```

```
Checked Bag'
    ELSE '10% off On One Hotel Meal'
    END AS perk_to_be_given
    FROM perks
    GROUP BY customer_segment
    ORDER BY customer_count
```

Annexe II : Tableau Dashboard Link

https://public.tableau.com/views/Book1_17194025097180/TravelTideDashboard?:language=en-US&publish=yes&:sid=&:redirect=auth&:display_count=n&:origin=viz_share_link