Mastery Project-

Reward Program Analysis

2024.06.18

Table of Contents

1. Project description

TravelTide, founded in April 2021, is a rapidly expanding online travel booking platform that offers users access to the largest travel inventory available. While TravelTide has concentrated on expanding its travel inventory and enhancing search capabilities, achieving a notable competitive edge in these areas, this narrow focus has resulted in the underdevelopment of other aspects of the customer experience. Consequently, the platform has faced challenges with low customer retention.

The objective of this project is to collaborate with Elena Tarrant to analyze the database, categorize the current user base, and design a corresponding rewards program tailored to these groups.

2. Project Summary

First, I conducted a preliminary exploration of the database using SQL. This initial step allowed me to understand the structure and content of the data. Following this, I undertook a thorough data cleaning process to ensure the accuracy and reliability of the dataset.

Next, in line with Elena Tarrant's requirements, I screened out the target group for research. This involved identifying and isolating the specific subset of users whose behaviors and characteristics aligned with the focus of the study.

I then proceeded to segment the user information at multiple levels. This detailed segmentation was performed using a combination of data-driven techniques and domain expertise. My goal was to create meaningful user groups that reflect distinct patterns and preferences.

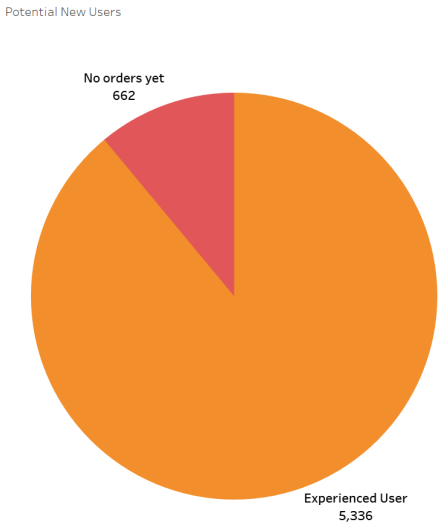
With the segmented data in hand, I carefully grouped the users based on both quantitative data analysis and qualitative insights. Each group was meticulously defined to ensure that the user characteristics were well represented.

Finally, I reviewed and verified the perks suggestions provided by Elena. This involved cross-referencing the proposed rewards with the user groups to ensure that the recommendations were appropriate and likely to enhance customer satisfaction and retention.

Throughout the process, my focus remained on creating a comprehensive and effective rewards program tailored to the diverse needs of TravelTide's user base.

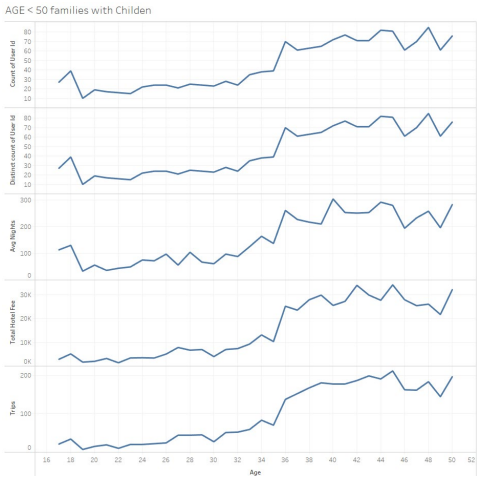
Group 1: Potential New Users: 10% Discount on First Order

Accounting for 12.4% of the active users on the website, these users have a good understanding of our platform but may lack the right opportunity to make a purchase. Offering a 10% discount on their first order could provide the perfect incentive to make their first booking.



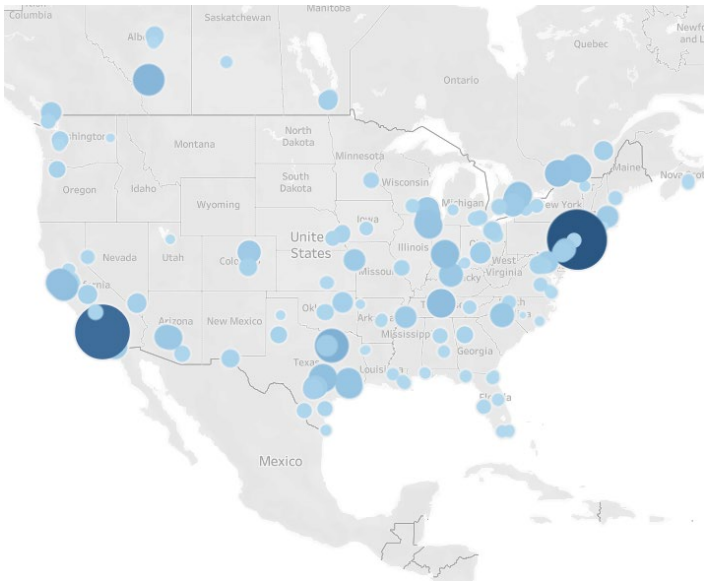
Group 2: Families with Children (AGE < 50): Free Hotel Meals

As children grow, families often look to travel to broaden their horizons. Families with children, where at least two people are traveling, will receive free meals at hotels. This perk aims to reduce the financial burden on parents and encourage family travel.



Group 3: Economic Hotspots: Exclusive Discounts

Focusing on New York and Los Angeles, these cities have the highest number of company orders and are hubs for corporate elites who value high quality and innovation. Users can exchange accumulated consumption for additional discounts, continuing to attract their business.



Group 4: Adults Under 50 without Children: No Cancellation Fees

This group is the most widely distributed and influential, with a high cancellation rate of 41.6%. Offering free cancellation can encourage them to explore various options without the fear of penalties, potentially increasing bookings.

Group 5: Adults 50 and Older: One Free Hotel Night with Flight Booking

Users in this age group tend to have stable consumption patterns and a slower pace of life. Providing a free hotel night near the airport with their flight booking can enhance their travel experience, offering convenience and comfort at the start of their trip.

CSV Link: https://docs.google.com/spreadsheets/d/1D-VTHCDuEWS-VFsoUe8o5dtEY9YseHNlDofU1iPR7aE/edit?usp=sharing

Appendix with SQLs:

```
-- Uessions starting after 2023-01-04
WITH
 sessions_20130104 AS (
  SELECT
   *
  FROM
   sessions s
  WHERE
   session_start >= '2023-01-04'
 ),
 -- Users with more than 7 sessions during the same time period
 users_8sessions AS (
  SELECT
   user_id,
   COUNT(*) AS session_count
  FROM
   sessions_20130104 s2
  GROUP BY
   user_id
  HAVING
   COUNT(DISTINCT s2.session_id) > 7
 ),
 -- Obtain the database designated by Elena
 data_filter AS (
  SELECT
   s2.*,
   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,
```

```
    ABS(h.nights) AS nights, -- Fix the problem of nights being negative
    h.rooms,
    h.check_in_time,
    h.check_out_time,
    h.hotel_per_room_usd
   FROM
    sessions_20130104 as s2
    LEFT JOIN users as u ON u.user_id = s2.user_id
    LEFT JOIN flights as f ON f.trip_id = s2.trip_id
    LEFT JOIN hotels as h ON h.trip_id = s2.trip_id
   WHERE
    s2.user_id IN (
     SELECT
      user_id
     FROM
      users_8sessions
    )
 ),
 -- Get the trip_ids that appear repeatedly and contain both false and true cancellation
states
 cancel_trip AS (
                    SELECT DISTINCT trip_id
                    FROM data_filter
   WHERE cancellation = true
          ),
 problem_cancel_session AS (
  SELECT session_id
  FROM data_filter
  WHERE cancellation = false
  AND trip_id IN (SELECT trip_id FROM cancel_trip)
 ),
 travel_tide_data AS (
  SELECT
   *,
  -- Calculate one-way flight distance
   2 * 6371 * ASIN(
    SQRT(
     POWER(
      SIN(
       RADIANS(df.destination_airport_lat - df.home_airport_lat) / 2
      ),
      2
     ) + COS(RADIANS(df.home_airport_lat)) * COS(RADIANS(df.destination_airport_lat)) *
POWER(
      SIN(
       RADIANS(df.destination_airport_lon - df.home_airport_lon) / 2
      ),
      2
     )
    )
   ) AS flight_distance_km
```

```
  FROM
    data_filter df
  WHERE
    session_id NOT IN (
      SELECT
        session_id
      FROM
        problem_cancel_session
    )
),
-- Basic information of the user
 user_info AS (
  SELECT DISTINCT
    user_id,
    EXTRACT(
      YEAR
      FROM
        AGE (CURRENT_DATE, ttd.birthdate)
    ) AS age,
    ttd.gender,
    ttd.married,
    ttd.has_children,
    ttd.home_country,
          ttd.home_city,
          ttd.home_airport_lat,
          ttd.home_airport_lon
  FROM
    travel_tide_data AS ttd
  GROUP BY
    user_id,
    EXTRACT(
      YEAR
      FROM
        AGE (CURRENT_DATE, ttd.birthdate)
    ),
    ttd.gender,
    ttd.married,
    ttd.has_children,
          ttd.home_country,
          ttd.home_city,
          ttd.home_airport_lat,
          ttd.home_airport_lon
 ),

 --User web page operation statistics
 user_sessions AS (
  SELECT DISTINCT
    user_id,
    COUNT(DISTINCT session_id) AS sessions,
    SUM(page_clicks) AS clicks
  FROM
```

```
  data_filter AS df
 GROUP BY DISTINCT
  user_id
),
--The user's valid flight reservation information
user_flight_booked AS (
 SELECT DISTINCT
  ttd.user_id,
  COUNT(DISTINCT ttd.trip_airline) AS flight_times,
  CEILING(AVG(ttd.checked_bags)) avg_checked_bag,
  ROUND(
   SUM(
    CASE
     WHEN return_flight_booked = TRUE THEN flight_distance_km * 2
     ELSE flight_distance_km
    END
   )::numeric,
   3
  ) AS total_flight_distance_km,
       COUNT(DISTINCT flight_discount_amount) AS use_flight_discount_times,
       ROUND(
   SUM(
    CASE
     WHEN flight_discount = TRUE THEN base_fare_usd * flight_discount_amount
     ELSE base_fare_usd
    END
   )::numeric,
   2
  ) AS total_flight_fee
 FROM
  travel_tide_data AS ttd
 WHERE
  ttd.trip_id IS NOT NUll
  AND ttd.flight_booked = true
       AND ttd.cancellation = false
 GROUP BY
  ttd.user_id
),
user_last_flight_booked AS (
 SELECT
  ttd.user_id,
  COUNT(*) AS last_minuten
 FROM
  travel_tide_data AS ttd
 WHERE
  session_start >= departure_time - INTERVAL '72 HOURS'
  AND ttd.trip_id IS NOT NUll
  AND ttd.flight_booked = true
       AND ttd.cancellation = false
 GROUP BY
  user_id
```

```
 ),
 user_non_America_flights AS (
  SELECT
   ttd.user_id,
   COUNT(*) AS user_non_America_flights
  FROM
   travel_tide_data AS ttd
  WHERE
   destination_airport_lon NOT BETWEEN -179.15 AND -52.62
   AND ttd.trip_id IS NOT NUll
   AND ttd.flight_booked = true
        AND ttd.cancellation = false
  GROUP BY
   user_id
 ),
 --The user's valid hotels reservation information
 user_hotel_booked AS (
  SELECT
        ttd.user_id,
   CEILING(AVG(ttd.nights)) avg_nights,
        CEILING(AVG(ttd.rooms)) avg_rooms,
        MAX(ttd.nights) max_nights,
  COUNT(DISTINCT hotel_discount_amount) AS use_hotel_discount_times,
        ROUND(
     SUM(
      CASE
       WHEN hotel_discount = TRUE THEN hotel_per_room_usd * hotel_discount_amount
       ELSE hotel_per_room_usd
      END
     )::numeric,
     2
   ) AS total_hotel_fee,
        COUNT(DISTINCT trip_id) AS trips,
        SUM(CASE WHEN flight_booked = TRUE THEN 1 END) AS flight_booking_count,
        SUM(CASE WHEN hotel_booked = TRUE THEN 1 END) AS room_booking_count

  FROM
   travel_tide_data AS ttd
  WHERE ttd.trip_id IS NOT NUll
   AND ttd.hotel_booked = true
        AND ttd.cancellation = false
  GROUP BY
        user_id
 )
--users_final_info AS (
SELECT ui.user_id,
(CASE
        WHEN trips IS NULL THEN 'Potential new users: 10% discount on first order.'
        WHEN AGE < 50  AND has_children = TRUE THEN 'AGE < 50 families with Childen:
free hotel meal'
```

```
        WHEN home_city IN ('new york', 'los angeles') THEN 'Economic hot spots: exclusive
disounts'
                WHEN AGE < 50 AND has_children = FALSE THEN 'AGE < 50 no child: no
cancellation fees'
                ELSE ' AGE >= 50 Group: 1 night free hotel with filght'
                END
  ) AS perk
FROM
  user_info AS ui
  LEFT JOIN user_sessions AS us ON us.user_id = ui.user_id
  LEFT JOIN user_flight_booked AS ufb ON ufb.user_id = ui.user_id
  LEFT JOIN user_last_flight_booked AS ulfb ON ulfb.user_id = ui.user_id
  LEFT JOIN user_non_America_flights AS unaf ON unaf.user_id = ui.user_id
  LEFT JOIN user_hotel_booked AS uhb ON uhb.user_id = ui.user_id
;
```