

Travel Tide Mastery Project



TravelTide

Esther Vollmer

November 2024
Masterschool DA May 24' Cohort

INTRODUCTION

Travel Tide serves a growing community of approximately 1 million users, generating over 5 million unique app sessions.

To encourage continued user engagement and loyalty, the company plans to introduce a rewards program tailored to user preferences and activity levels.

This initiative is being led by Elena, the Head of Marketing, in collaboration with the Data team. The Data team has been tasked with identifying active users and recommending optimal reward perks to ensure the program's success.

OBJECTIVE

Develop a customer segmentation process by leveraging data from a database through a series of analytical steps, including filtering, feature engineering, clustering, and in-depth analysis. Filtering ensures the data is relevant and clean, while feature engineering transforms raw data into meaningful variables that capture key customer characteristics. Clustering techniques are then used to group customers into segments based on similar behaviors, preferences, or attributes, enabling a more structured understanding of the customer base.

The ultimate goal of this segmentation is to gain actionable insights into customer behavior and design strategies to enhance their experience. By assigning tailored perks or incentives to each cluster based on its specific needs and preferences, businesses can foster stronger relationships with their customers. This personalized approach not only increases customer satisfaction but also boosts retention, ensuring long-term business success.

MATERIALS & PROCEDURE

SQL and Tableau were essential tools in the customer segmentation process. SQL facilitated efficient data extraction, cleaning, and transformation from the database, ensuring the preparation of accurate datasets for analysis. Tableau's powerful visualization capabilities enabled the exploration of cluster characteristics and patterns, presenting actionable insights through intuitive dashboards to support the creation of tailored customer perks.

DATA & SOLUTION

CATEGORY	PROPOSED PERKS
Elite Value Perks	<ol style="list-style-type: none"> 1. Priority Boarding (Time and comfort) 2. Additional Luggage Allowance (Extra bags for premium travelers) 3. Exclusive Airport Lounge Access (Comfort before flights) 4. Travel Insurance (Peace of mind for premium travelers) 5. Complimentary Upgrades (First Class/Business Class) (Luxury travel experience)
Premium Perks Economy	<ol style="list-style-type: none"> 1. Frequent Flyer Bonus (Fly far for higher points) (Rewards for repeat travelers) 2. No Additional Charges for Seat Selection (Personalized experience without extra cost) 3. Airport Shuttle Service (Within 10 km from Airport) (Convenient transport) 4. Travel Insurance (Basic coverage for frequent travelers) 5. 1 Free 7kg Checked-in Luggage (Practical and valuable for frequent travelers)

What insights was made to create the segmentation of clusters?

1. Session Frequency and Interaction

Segment users based on how often they engage with the app (daily, weekly, or monthly), how long their sessions last, and the specific features they use (e.g., browsing deals, booking, or redeeming rewards). This identifies core, occasional, and dormant users.

2. Behavioral Patterns and Preferences

Group users by their booking behavior, such as last-minute planners vs. advance bookers, solo travelers vs. families, or budget-conscious vs. premium spenders. Additionally, analyze their spending habits and engagement journey through the app to understand their travel preferences and purchase triggers.

3. Demographic Profiles and Localization

Segment users by demographic factors like age, lifestyle stage (e.g., family travelers, working professionals), and location (city, region, or country). This allows tailoring offers to suit local preferences and accessibility while addressing unique demographic needs.

4. Integrated Lifestyle-Based Personas

Combine session data, behavior, and demographics to create detailed personas. For example, “Frequent family travelers with short but frequent sessions,” or “Millennials seeking budget travel deals with high session engagement.” This approach enables highly personalized targeting and communication.

CONCLUSION OF STRATEGY

- A/B testing will help understand customer behavior, preferences, and pain points for both new and returning customers.
- The 6-12 month timeline allows for ample testing, refinement, and iteration based on real-time data, ensuring campaigns are continuously optimized.
- Creating 3 new campaigns will address customer needs and capitalize on the insights from the A/B tests, whether it’s solving a problem, promoting upsells, or acquiring new customers.
- The sales push will amplify these efforts with targeted, time-sensitive promotions designed to drive immediate sales and achieve specific business objectives.

By combining these elements, a comprehensive strategy is built that not only drives short-term results but also sets up TravelTide for sustainable growth over time.

APPENDIXES

Tableau Viz:

<https://public.tableau.com/app/profile/ea.viz/viz/TravelTideProject2024/CLUSTERING>

Github: <https://github.com/ea-techcodes/TravelTide-Project>

SQL Syntax:

-- Select users with more than 7 sessions after a specific date

WITH filtered_users AS (

SELECT

user_id

FROM

sessions

WHERE

session_start > '2023-01-04' -- Filter sessions starting after January 4, 2023

GROUP BY

user_id

HAVING

COUNT(session_id) > 7 -- Retain users with more than 7 sessions

),

-- Combine user, session, hotel, and flight data at the session level

session_level AS (

SELECT

u.user_id,

s.session_id,

s.trip_id,

u.birthdate,

EXTRACT(YEAR FROM AGE(u.birthdate)) AS age, -- Calculate user 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,
u.sign_up_date,
s.session_start,
s.session_end,
(EXTRACT(EPOCH FROM (s.session_end - s.session_start))) AS session_duration_in_seconds, --
Session duration
s.flight_discount,
s.hotel_discount,
s.flight_discount_amount,
s.hotel_discount_amount,
s.flight_booked,
s.hotel_booked,
s.page_clicks,
s.cancellation,
h.hotel_name,
-- Handle edge cases for hotel stay duration
CASE
    WHEN h.nights < 0 THEN ABS(h.nights)
    WHEN h.nights = 0 THEN 1
    ELSE h.nights
END AS nights,
h.rooms,
-- Handle inconsistent check-in and check-out times
CASE

```

```

        WHEN h.check_in_time > h.check_out_time THEN h.check_out_time
        ELSE h.check_in_time
    END AS check_in_time,
CASE
    WHEN h.check_out_time < h.check_in_time THEN h.check_in_time
    ELSE h.check_out_time
END AS check_out_time,
h.hotel_per_room_usd,
f.origin_airport,
f.destination,
f.destination_airport,
f.seats,
f.return_flight_booked,
f.departure_time,
EXTRACT(MONTH FROM departure_time) AS departure_month, -- Extract departure month
f.return_time,
f.checked_bags,
f.trip_airline,
f.destination_airport_lat,
f.destination_airport_lon,
f.base_fare_usd
FROM
    filtered_users AS fs
JOIN
    users AS u ON fs.user_id = u.user_id
LEFT JOIN
    sessions AS s ON s.user_id = fs.user_id
LEFT JOIN
    hotels AS h ON s.trip_id = h.trip_id

```

```

LEFT JOIN
    flights AS f ON s.trip_id = f.trip_id
ORDER BY
    u.user_id ASC
),

-- Aggregate session-level data to derive trip-level insights
trip_level AS (
    SELECT
        user_id,
        COUNT(trip_id) AS total_trips, -- Total trips per user
        SUM(CASE
            WHEN flight_booked AND return_flight_booked THEN 2 -- Round trips count as 2 flights
            WHEN flight_booked THEN 1 -- One-way flight
            ELSE 0
        END) AS total_flights,
        COUNT(hotel_booked) FILTER (WHERE hotel_booked = 'true') AS total_hotel_booked, -- Total hotel
bookings
        SUM((hotel_per_room_usd * nights * rooms) * (1 - COALESCE(hotel_discount_amount, 0))) AS
money_spent_hotel, -- Hotel spending
        SUM((base_fare_usd) * (1 - COALESCE(flight_discount_amount, 0))) AS money_spent_flight, --
Flight spending
        SUM((hotel_per_room_usd * nights * rooms) * (1 - COALESCE(hotel_discount_amount, 0))) +
        SUM((base_fare_usd) * (1 - COALESCE(flight_discount_amount, 0))) AS money_spent_booking, --
Total spending
        SUM(EXTRACT(DAY FROM departure_time - session_end)) AS total_time_before_trip, -- Time
between booking and trip
        SUM(haversine_distance(home_airport_lat, home_airport_lon, destination_airport_lat,
destination_airport_lon)) AS km_flow -- Total distance flown
    FROM
        session_level
    WHERE

```



```

trip_id IS NOT NULL -- Exclude non-trip sessions

AND trip_id NOT IN (SELECT DISTINCT trip_id FROM session_level WHERE cancellation) --
Exclude canceled trips

GROUP BY

    user_id

),

-- Aggregate session and trip data at the user level
user_level AS (
    SELECT
        user_id,
        age,
        gender,
        married,
        has_children,
        home_country,
        home_city,
        home_airport,
        home_airport_lat,
        home_airport_lon,
        sign_up_date,
        SUM(page_clicks) AS total_clicks, -- Total clicks across all sessions
        SUM(nights) AS total_nights, -- Total nights booked
        SUM(rooms) AS total_rooms, -- Total rooms booked
        ROUND(AVG(hotel_per_room_usd), 2) AS avg_hotel_per_room_usd, -- Average hotel cost
        ROUND(AVG(base_fare_usd), 2) AS avg_base_fare_usd, -- Average flight fare
        COUNT(DISTINCT session_id) AS session_count, -- Total sessions
        ROUND(AVG(session_duration_in_seconds), 0) AS avg_session_duration_in_seconds, -- Average
session duration
        COUNT(cancellation) FILTER (WHERE cancellation = 'true') AS total_cancellation, -- Total
cancellations

```

```

COUNT(flight_discount) FILTER (WHERE flight_discount = 'true') AS total_flight_with_discount, --
Flights with discounts

COUNT(hotel_discount) FILTER (WHERE hotel_discount = 'true') AS total_hotel_with_discount, --
Hotels with discounts

ROUND(AVG(flight_discount_amount), 2) AS avg_flight_discount, -- Average flight discount

ROUND(AVG(hotel_discount_amount), 2) AS avg_hotel_discount, -- Average hotel discount

-- Conversion rates

ROUND(COUNT(flight_booked) FILTER (WHERE flight_booked = 'true') / COUNT(DISTINCT
session_id)::NUMERIC , 2) AS con_rate_flights,

ROUND(COUNT(hotel_booked) FILTER (WHERE hotel_booked = 'true') / COUNT(DISTINCT
session_id)::NUMERIC , 2) AS con_rate_hotels,

ROUND((COUNT(flight_booked) FILTER (WHERE flight_booked = 'true') + COUNT(hotel_booked)
FILTER (WHERE hotel_booked = 'true')) / COUNT(DISTINCT session_id)::NUMERIC , 2) AS
con_rate_combined,

COUNT(return_flight_booked) FILTER (WHERE return_flight_booked = 'true') AS
total_return_flight_booked, -- Total return flights

-- User segmentation

CASE

    WHEN age BETWEEN 17 AND 25 THEN '17-25'

    WHEN age BETWEEN 26 AND 35 THEN '26-35'

    WHEN age BETWEEN 36 AND 50 THEN '36-50'

    ELSE '50+'

END AS age_bucket,

CASE

    WHEN married = 'true' AND has_children = 'true' THEN 'Married With Children'

    WHEN married = 'true' AND has_children = 'false' THEN 'Married With No Children'

    WHEN married = 'false' AND has_children = 'true' THEN 'Single With Children'

    WHEN married = 'false' AND has_children = 'false' THEN 'Single With No Children'

    ELSE 'Unknown'

END AS family_status,

-- Additional segmentations

CASE

```

```

    WHEN SUM(page_clicks) BETWEEN 7 AND 20 THEN 'Low Activity'

    WHEN SUM(page_clicks) BETWEEN 21 AND 66 THEN 'Medium Activity'

    WHEN SUM(page_clicks) > 66 THEN 'High Activity'

    ELSE 'Unknown'

END AS activity_level,

-- Cancellation behavior based on rates

CASE

    WHEN COUNT(cancellation) FILTER (WHERE cancellation = 'true') / NULLIF(COUNT(trip_id), 0)
<= 0.1 THEN 'Low Cancellation'

    WHEN COUNT(cancellation) FILTER (WHERE cancellation = 'true') / NULLIF(COUNT(trip_id), 0)
BETWEEN 0.1 AND 0.3 THEN 'Medium Cancellation'

    WHEN COUNT(cancellation) FILTER (WHERE cancellation = 'true') / NULLIF(COUNT(trip_id), 0)
> 0.3 THEN 'High Cancellation'

    ELSE 'No Flights Booked'

END AS cancellation_behavior,

-- Discount usage behavior

CASE

    WHEN (COUNT(flight_discount) FILTER (WHERE flight_discount = 'true') +
COUNT(hotel_discount) FILTER (WHERE hotel_discount = 'true')) >= 5 THEN 'Frequent Discount User'

    WHEN (COUNT(flight_discount) FILTER (WHERE flight_discount = 'true') +
COUNT(hotel_discount) FILTER (WHERE hotel_discount = 'true')) BETWEEN 1 AND 4 THEN
'Occasional Discount User'

    WHEN (COUNT(flight_discount) FILTER (WHERE flight_discount = 'true') +
COUNT(hotel_discount) FILTER (WHERE hotel_discount = 'true')) = 0 THEN 'Non-Discount User'

    ELSE 'Unknown'

END AS discount_usage_behavior,

-- Booking preferences

CASE

    WHEN COUNT(hotel_booked) FILTER (WHERE hotel_booked = 'true') = 0 AND
COUNT(flight_booked) FILTER (WHERE flight_booked = 'true') > 0 THEN 'Flight Only'

    WHEN COUNT(flight_booked) FILTER (WHERE flight_booked = 'true') = 0 AND
COUNT(hotel_booked) FILTER (WHERE hotel_booked = 'true') > 0 THEN 'Hotel Only'

    WHEN COUNT(flight_booked) FILTER (WHERE flight_booked = 'true') > 0 AND
COUNT(hotel_booked) FILTER (WHERE hotel_booked = 'true') > 0 THEN 'Both Flight and Hotel'

```

```

        ELSE 'No Bookings'
    END AS booking_type_preference,

    -- Interaction duration categorization

    CASE

        WHEN ROUND(AVG(session_duration_in_seconds), 0) <= 230 THEN 'Short Sessions'

        WHEN ROUND(AVG(session_duration_in_seconds), 0) BETWEEN 231 AND 1200 THEN
'Moderate Sessions'

        WHEN ROUND(AVG(session_duration_in_seconds), 0) > 1200 THEN 'Long Sessions'

        ELSE 'Unknown'

    END AS interaction_duration,

    -- Travel frequency segmentation

    CASE

        WHEN (COUNT(flight_booked) FILTER (WHERE flight_booked = 'true') + COUNT(hotel_booked)
FILTER (WHERE hotel_booked = 'true')) = 0 THEN 'No Bookings'

        WHEN (COUNT(flight_booked) FILTER (WHERE flight_booked = 'true') + COUNT(hotel_booked)
FILTER (WHERE hotel_booked = 'true')) BETWEEN 1 AND 2 THEN 'Infrequent Traveler'

        WHEN (COUNT(flight_booked) FILTER (WHERE flight_booked = 'true') + COUNT(hotel_booked)
FILTER (WHERE hotel_booked = 'true')) BETWEEN 3 AND 5 THEN 'Occasional Traveler'

        WHEN (COUNT(flight_booked) FILTER (WHERE flight_booked = 'true') + COUNT(hotel_booked)
FILTER (WHERE hotel_booked = 'true')) >= 6 THEN 'Frequent Traveler'

    END AS loyalty_level,

    -- Hotel price preferences

    CASE

        WHEN ROUND(AVG(hotel_per_room_usd), 2) <= 100 THEN 'Budget'

        WHEN ROUND(AVG(hotel_per_room_usd), 2) BETWEEN 101 AND 300 THEN 'Mid-Range'

        WHEN ROUND(AVG(hotel_per_room_usd), 2) > 300 THEN 'Luxury'

        ELSE 'Unknown'

    END AS hotel_price_category,

    -- Flight fare preferences

    CASE

        WHEN ROUND(AVG(base_fare_usd), 2) <= 200 THEN 'Budget'

        WHEN ROUND(AVG(base_fare_usd), 2) BETWEEN 201 AND 1000 THEN 'Mid-Range'

```

```

        WHEN ROUND(AVG(base_fare_usd), 2) > 1000 THEN 'Luxury'
        ELSE 'Unknown'
    END AS flight_fare_category
FROM
    session_level
GROUP BY
    user_id, age, gender, married, has_children, home_country, home_city, home_airport,
    home_airport_lat, home_airport_lon, sign_up_date, age_bucket, family_status
)

```

-- Combine user-level and trip-level data

```

SELECT
    ul.* -- Include all user-level metrics
    tl.* -- Include all trip-level metrics
FROM
    user_level AS ul
LEFT JOIN
    trip_level AS tl
ON ul.user_id = tl.user_id

```