

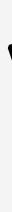


Namma Yatri Data Analysis

Report

-By Abhishek Kumar Singh



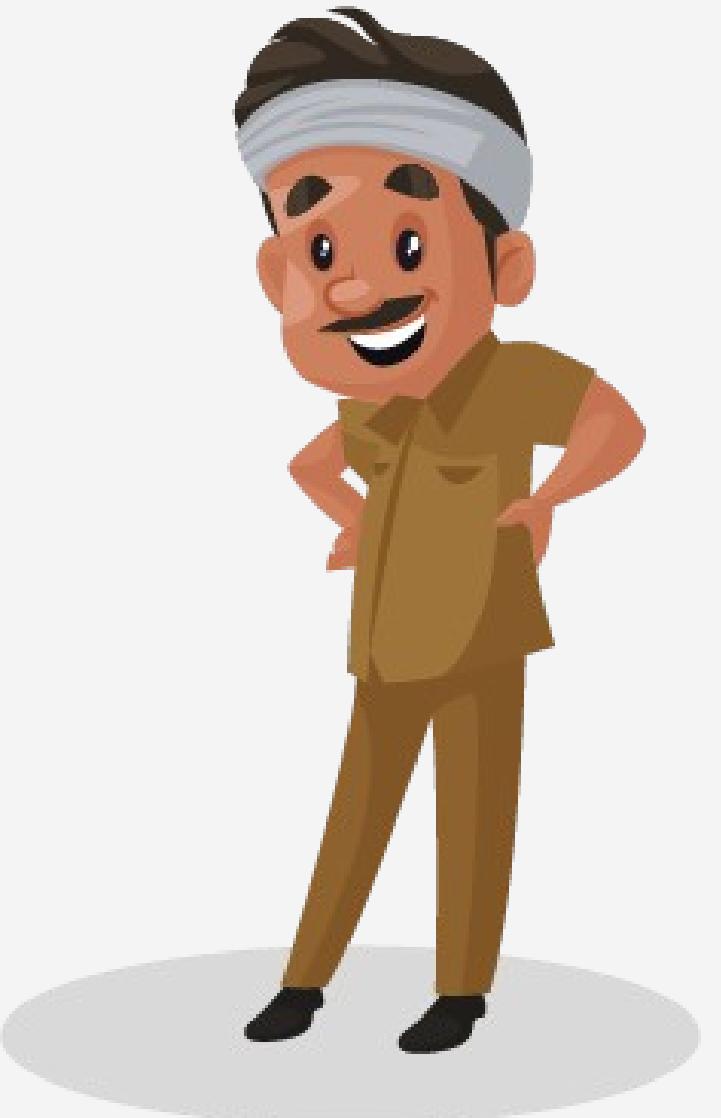


PROJECT DESCRIPTION

Namma Yatri is a ride-hailing app that facilitates transportation services by connecting drivers and customers.

This analysis aims to extract insights from trip data, including total trips, earnings, cancellations, searches, and payment methods, to improve user experience and operational efficiency.

By examining key metrics such as trip searches, fare estimates, driver quotes, cancellations, payments, and completed trips, the goal is to uncover trends and identify areas for improvement.

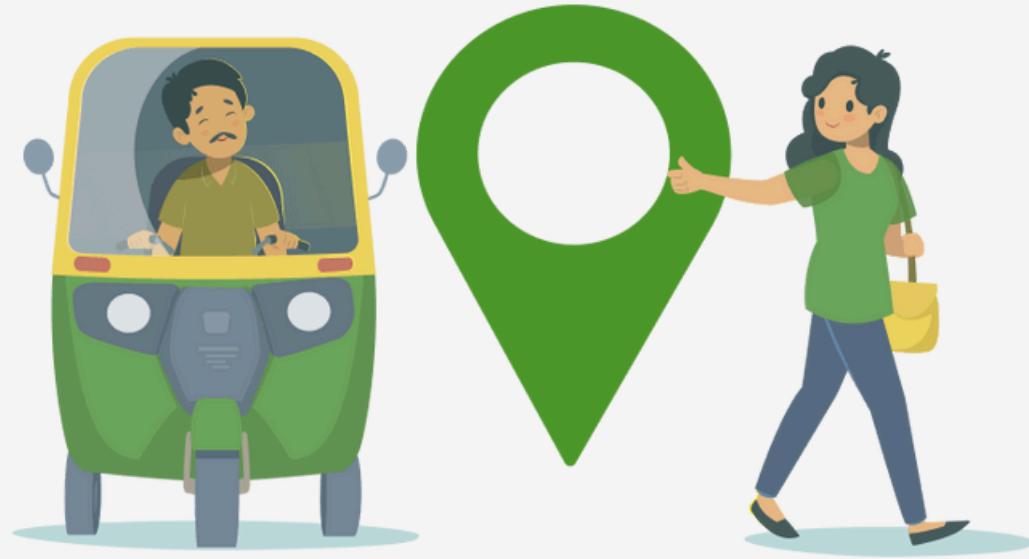


PROBLEM STATEMENT

Namma Yatri faces challenges in ride completion, cancellations, and operational efficiency. Many users drop off before completing trips, and both drivers and customers frequently cancel rides, impacting revenue. Understanding driver availability, pricing impact, and payment preferences is crucial.

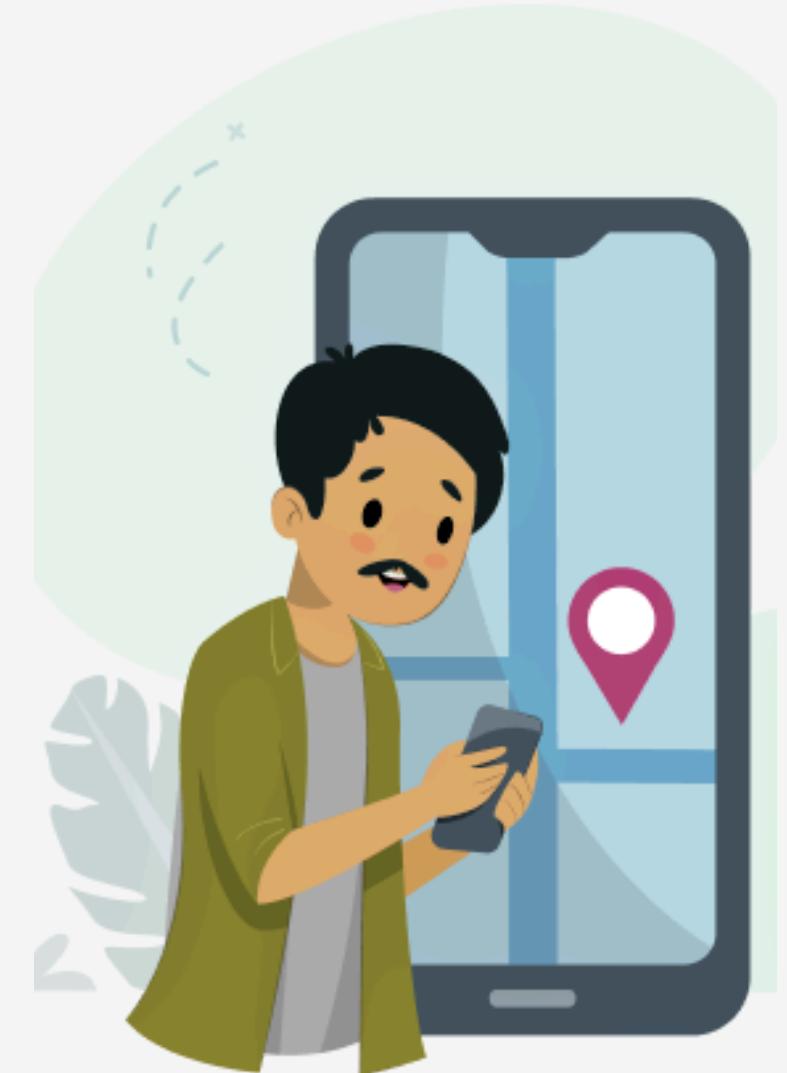
Location and time-based demand variations affect service efficiency, while trip distance, duration, and waiting times influence customer satisfaction.

This analysis aims to identify key issues and provide data-driven solutions to improve ride completion rates, reduce cancellations, and optimize platform performance.



DATA COLLECTION & UNDERSTANDING

- The app collects data on user interactions, such as searches, fare estimates, ride requests, and cancellations.
- The trip details dataset contains binary values (1 for YES, 0 for NO) for key actions like fare estimates, ride requests, driver allocation, OTP entry, and ride completion.
- Payment methods, trip durations, and location trends are also recorded.





nammayatri

TRIPS DASHBOARD

Fare Method

All

Select Assembly

All

Completed Trips

983

Searches

2161

Estimates

1758

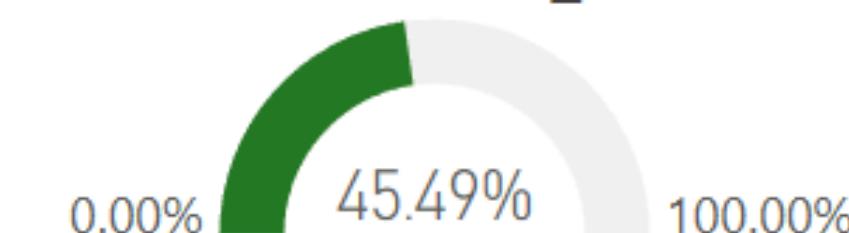
Quotes

1277

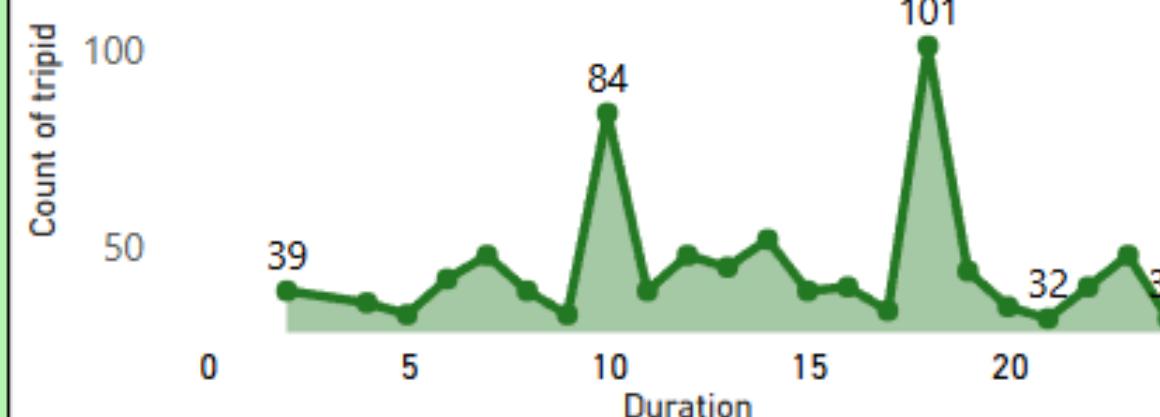
Driver Earnings

751K

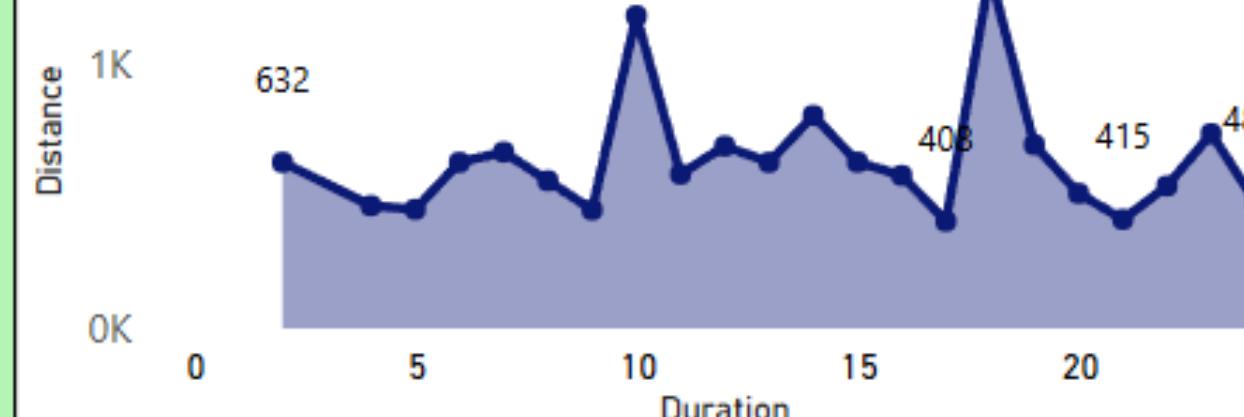
Conversion_rate



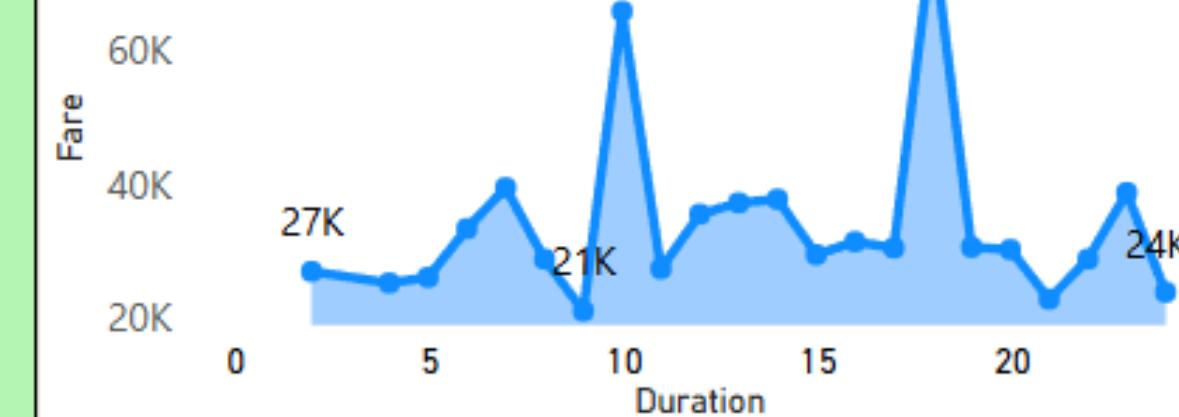
Trips Vs Duration



Distance Vs Duration

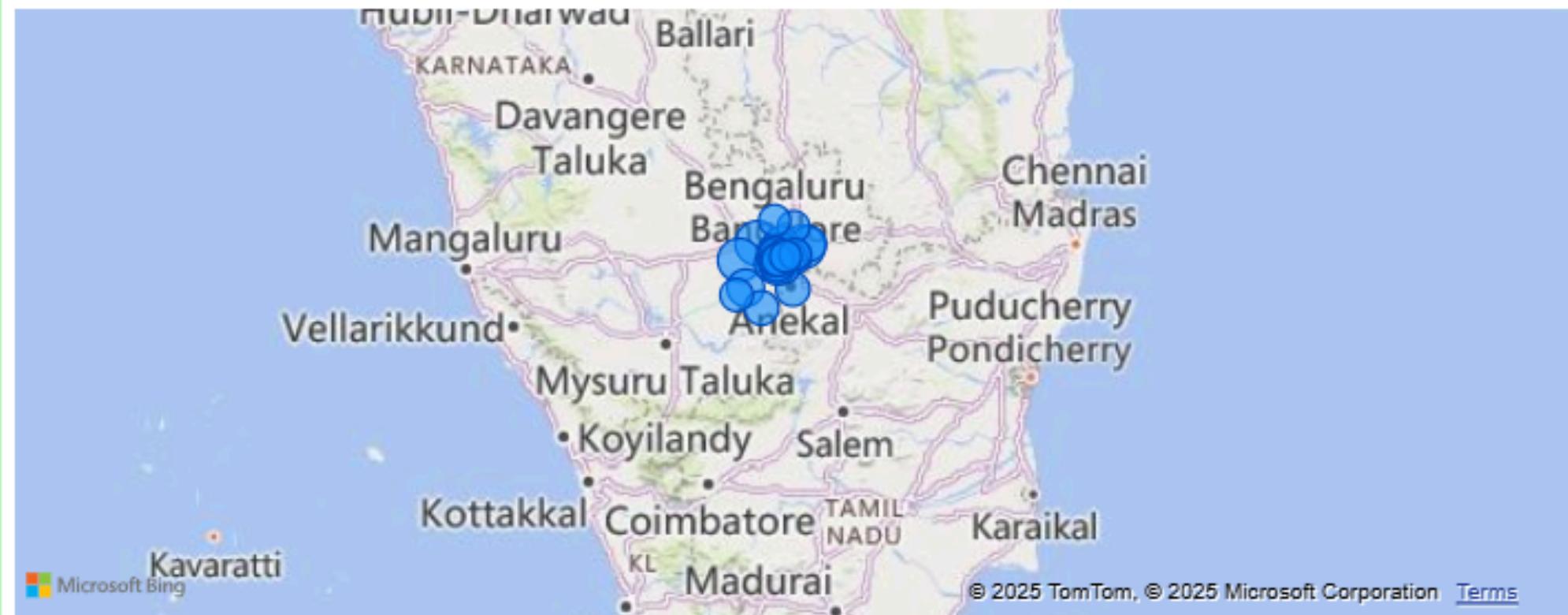


Fare Vs Duration



Assembly	Total_Searches	Total_searches_for_quotes	Total_searches_got_earned
Anekal	60	40	48
B. T. M. Layout	56	37	44
Bangalore South	57	47	51
Basavanagudi	59	38	46
Bommanahalli	58	43	49
Byatarayanapura	53	34	45
C. V. Raman Nagar	64	42	52
Chamrajpet	53	39	43
Chennapetra	56	40	44
Total	2161	1455	

Count of tripid by assembly.Assembly





KEY METRICS AND FINDINGS



Overall Trip Statistics

- Total Trips: 983
- Total Unique Trip IDs: 2161
- Total Drivers: 30
- Total Earnings: ₹751,343
- Total Completed Trips: 983



Search and Booking Funnel

Stage	Count	Conversion Rate
Total Searches	2161	100%
Searches Got Estimate	1758	81%
Searches for Quotes	1455	83% of estimated searches
Searches Got Quotes	1277	88% of quote searches
OTP Entered	983	77% of quoted searches
Ride Completed	983	100% of OTP entered

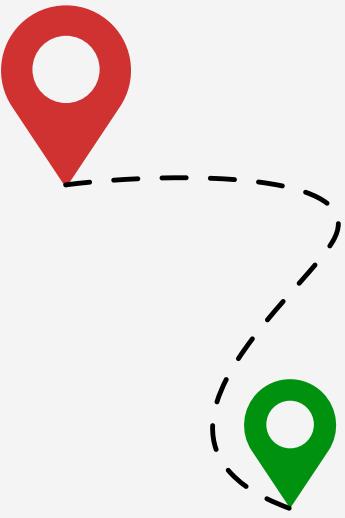
KEY METRICS AND FINDINGS

Cancellations Analysis

- Trips Cancelled by Drivers: 1021
- Trips Cancelled by Customers: 1041
- Highest Driver Cancellations by Location: Location ID 1 (43 cancellations)
- Highest Customer Cancellations by Location: Location ID 4 (40 cancellations)

Distance and Fare Analysis

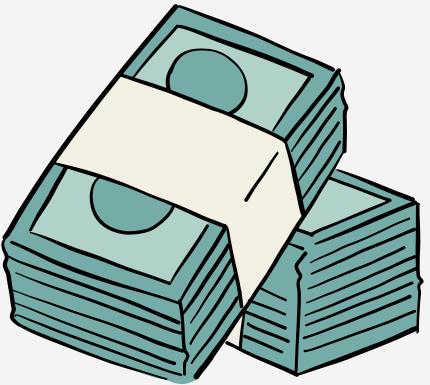
- Average Distance per Trip: 14.39 km
- Average Fare per Trip: ₹764.34
- Total Distance Traveled: 14,148 km
- Highest Earning Location: Location ID 6 (₹30,295 total earnings)
- Highest Earning Duration: Hour 18 (₹77,560 total earnings)



KEY METRICS AND FINDINGS

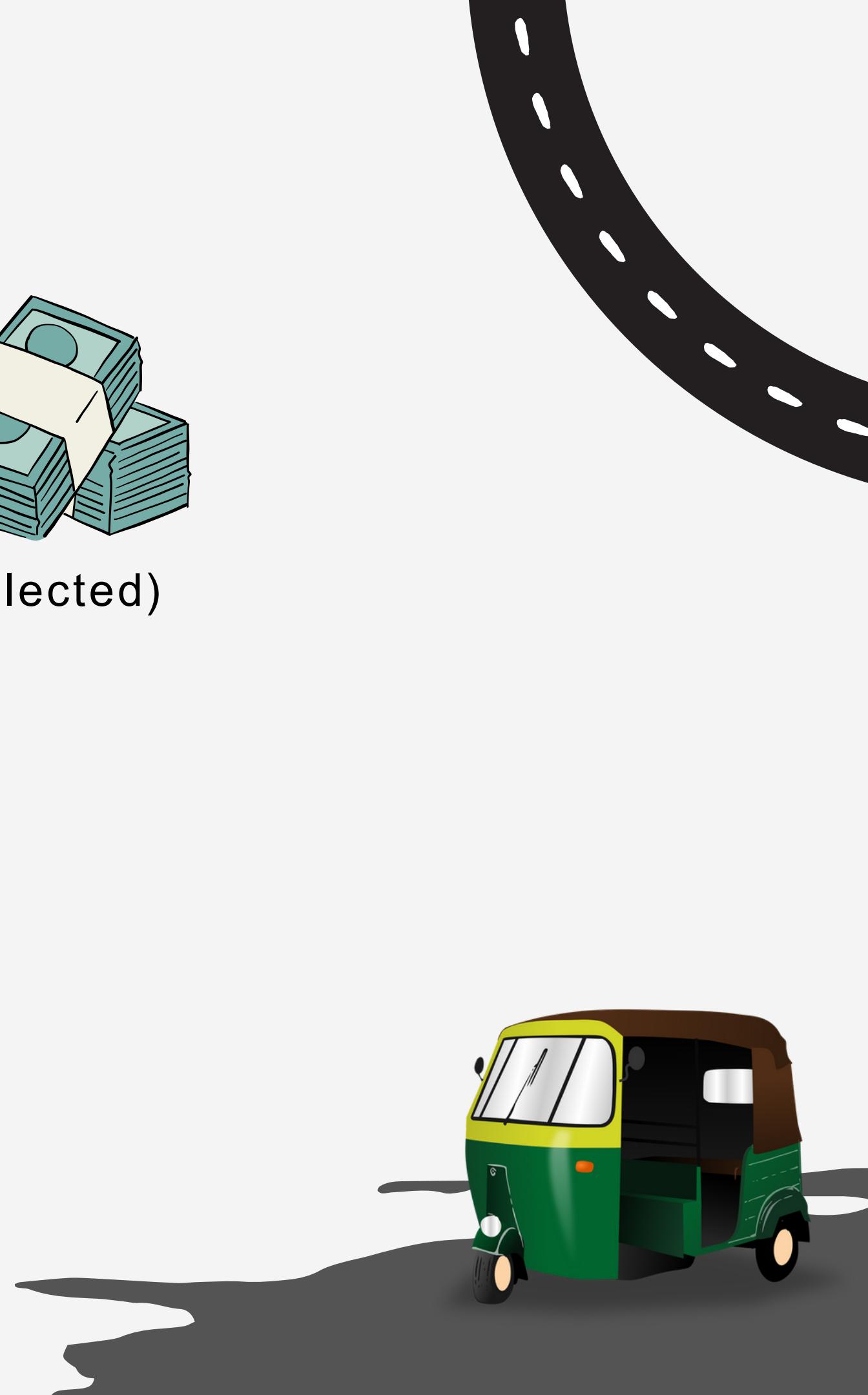
Payment Method Trends

- Most Used Payment Method: UPI (505 transactions)
- Cash Payments: 478 transactions
- Highest Payment Made Through: UPI (highest total fare collected)



Location-Based Trip Trends

- Top 2 Locations with Most Trips:
 - Nelamangala: 113 trips
 - Hoskote: 104 trips



KEY METRICS AND FINDINGS

Driver Performance Analysis

- Top 5 Earning Drivers:
 - a. Driver ID 12 (₹36,787)
 - b. Driver ID 8 (₹30,101)
 - c. Driver ID 21 (₹29,787)
 - d. Driver ID 24 (₹28,870)
 - e. Driver ID 30 (₹28,853)



- **Driver-Customer Pair with Most Rides:**
 - Driver ID 17, Customer ID 96 (4 rides)
 - Driver ID 28, Customer ID 15 (4 rides)



PROBLEMS AND POSSIBLE SOLUTIONS

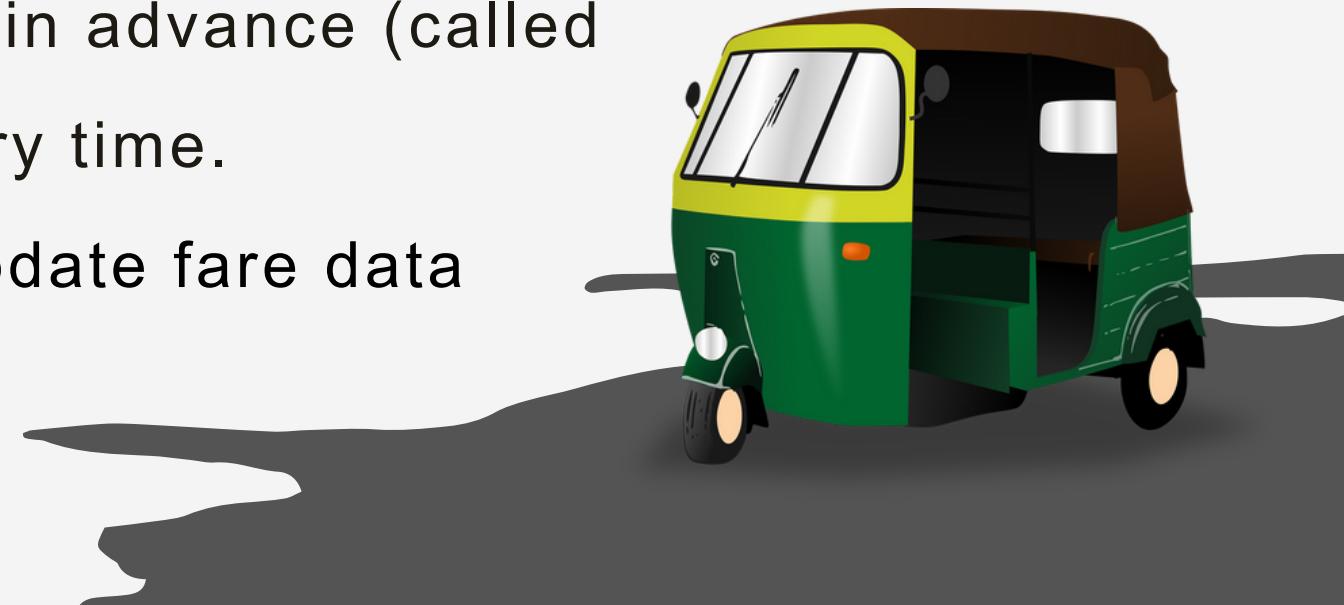
“Searches → Got Fare Estimate (81% conversion, 19% drop-off)”

Possible Reasons

- Users might experience slow responses due to server-side lag or front-end rendering issues.
- Users may abandon the session if fare loading takes too long or they get distracted.
- Incorrect or incomplete pickup/drop locations can block fare generation.
- Newly added locations or rare route combinations may not have fare data.
- Weak signal or offline mode can prevent fare data from loading.

Possible Solutions

- **Make the backend faster:** Save fare details for common routes in advance (called caching), so the app can show fares quickly without waiting every time.
- **Expand Fare Coverage Across Routes:** Regularly audit and update fare data for all common and newly added areas.





PROBLEMS AND POSSIBLE SOLUTIONS

Got Fare Estimate → Searched for Quotes (83%) Drop-off: 17%

Users saw fare but didn't proceed to request quotes

Possible Reasons

- Fare too high: Users may feel the fare isn't worth it.
- Just checking prices (no intent to ride)
- No offers/discounts
- Better deals on Ola/Uber
- Preferred to choose any near by vehicle.

Possible Solutions

- Introduce fare comparison widget: “cheaper than other platforms”
- Display trip benefits (e.g., no surge, reliable drivers).





PROBLEMS AND POSSIBLE SOLUTIONS

Request Ride → Got Driver Quote (88%) Drop-off: 12%

User requested quotes but didn't receive any drivers.

Possible Reasons

- Low driver availability in that area or time.
- High driver rejection rate for certain routes (1021 rides cancelled by drivers).
- Odd routes or short-distance rides not profitable

Possible Solutions

- Prioritize matching drivers with a low cancellation history.
- Provide incentives for drivers to accept and complete trips.





PROBLEMS AND POSSIBLE SOLUTIONS

Got Quote → OTP Entered (77%) Drop-off: 23%

Trip accepted but OTP was never entered.

Possible Reasons

- Customer canceled: Driver rating or vehicle type not preferred.
- Driver didn't like destination or fare.
- Driver too far away from pickup.
- Driver facing personal issue or multiple bookings.

Possible Solutions

- Penalize frequent driver cancellations.
- Let drivers preview trip before accepting.
- Add bonus/incentives for completing certain trips.





PROBLEMS AND POSSIBLE SOLUTIONS

OTP Entered → Ride Completed (100%)

Whenever OTP was entered, the trip was always completed — shows strong reliability once ride starts.



Namma Yatri Analysis Sql Queries

1. total trips

```
select count(*)as completed_trips from trips; #983  
select count(distinct tripid) as all_trips from trips_details4; #2161
```

2. total drivers

```
select count(distinct driverid) as total_drivers from trips; #30
```

3. total earnings

```
select sum(fare) as total_earnings from trips; # 751343
```

4. total Completed trips

```
select count(tripid) from trips_details4 where end_ride =1; #983
```

5. total searches

```
select sum(searches) as searches from trips_details4; #2161
```

6. total searches which got estimate

```
select sum(searches_got_estimate) as searches from trips_details4; #1758
```

7. total searches for quotes

```
select sum(searches_for_quotes) as searches from trips_details4; #1455
```

8. total searches which got quotes

```
select sum(searches_got_quotes) as searches from trips_details4; #1277
```

9. trips cancelled by drivers

```
select count(driver_not_cancelled) as cancelled_by_drivers  
from trips_details4 where driver_not_cancelled=0; #1021
```

10. total otp entered

```
select sum(otp_entered) as total_otp_entered from trips_details4; #983
```

11. total end ride

```
select sum(end_ride) as total_end_ride from trips_details4;#983
```

12. average distance per trip

```
select avg(distance) as average_distance_per_trip from trips;#14.3927
```

13. average fair per trip

```
select avg(fare) as average_fair_per_trip from trips;#764.3367
```

14. distance travelled

```
select sum(distance) from trips;#14148
```

15. most used payment method

```
select method, count(faremethod)  
from trips join payment on  
trips.faremethod = payment.id  
group by method order by count(faremethod) desc ;  
#upi    505  
#cash   478
```

16. the highest payment was made through which instrument

```
with cte as (  
    select faremethod, sum(fare) as total_fare from trips  
    group by faremethod  
)  
select p.method  
from cte join payment p on p.id = cte.faremethod  
order by cte.total_fare desc limit 1; #upi
```

17. which two location had the most trips

```
select assembly1, count(t.loc_from) as total_trips from
trips_details4 as t
join loc on
t.loc_from = loc.id
group by assembly1 order by total_trips desc limit 2;
#Nelamangala    113
#Hoskote      104
```

18. top 5 earning drivers

```
select * from
(select *,dense_rank() over(order by fare desc) rnk from
(select driverid,sum(fare) fare from trips
group by driverid) as b) as c where rnk<6 ;
-- driverid fare rnk
-- 12    36787    1
-- 8     30101    2
-- 21    29787    3
-- 24    28870    4
-- 30    28853    5
```

19. which duration had most trips

```
select * from
(select *,rank() over(order by cnt desc)rnk from
(select duration, count(distinct tripid)cnt
from trips group by duration) b) c where rnk=1;
duration  cnt  |  rnk
18        101   1
```

20. which driver, customer pair had more orders

```
select * from
(select *, dense_rank() over (order by cnt desc) as ran_k from
(select driverid,custid,count(distinct tripid) as cnt from trips
group by driverid,custid) a) c  where ran_k=1;
driverid  custid |  cnt  |  ran_k
17        96      4      1
28        15      4      1
```

21. which area got the highest trips in which duration

```
select * from (
  select *, rank() over(partition by duration order by cnt desc) as rnk from
  (select duration,loc_from, count(distinct tripid) as cnt from trips
  group by duration,loc_from)a )b
where rnk=1;
```

duration	loc_from	cnt	rnk
2	9	5	1
4	4	3	1
5	12	3	1
5	26	3	1
5	36	3	1
6	9	4	1
7	19	7	1
8	4	3	1
8	24	3	1
8	26	3	1
8	35	3	1
9	35	4	1
10	12	5	1

22. area got the highest fares

```
select * from
(select *,rank()over(order by fare desc) rnk
from
(select loc_from, sum(fare) fare from trips
group by loc_from)b)c
where rnk=1;
```

loc_from	fare	rnk
6	30295	1

23. area got the highest DRIVER cancellations

```
select * from
(select *,rank()over(order by cancel desc) rnk
from
(select loc_from, count(*) -sum(driver_not_cancelled) cancel
from trips_details4 group by loc_from )b)c where rnk=1;
```

loc_from	cancel	rnk
1	43	1

24. area got the highest CUSTOMER cancellations

```
select * from
(select *,rank()over(order by cancel desc) rnk
from
(select loc_from, count(*) -sum(customer_not_cancelled) cancel
from trips_details4 group by loc_from )b)c where rnk=1;
| loc_from | cancel | rnk |
| 4        |    40   | 1  |
```

25. which duration got highest trips and fairs

```
select * from
(select *,rank()over(order by fare desc) rnk
from
(select duration,sum(fare) fare from trips group by duration)b)c where rnk=1;
| duration | fare | rnk |
| 18       | 77560 | 1  |
```