

---

---

# Metrocar

## Аналіз клієнтської воронки

Катерина Майкова • 01.06.2025

---

# Мета

Дослідити, чому багато користувачів не доходять до першої завершеної поїздки.  
Як оптимізувати досвід користувача та покращити конверсію на кожному етапі.  
Розробити [дашборд](#) клієнтської воронки.

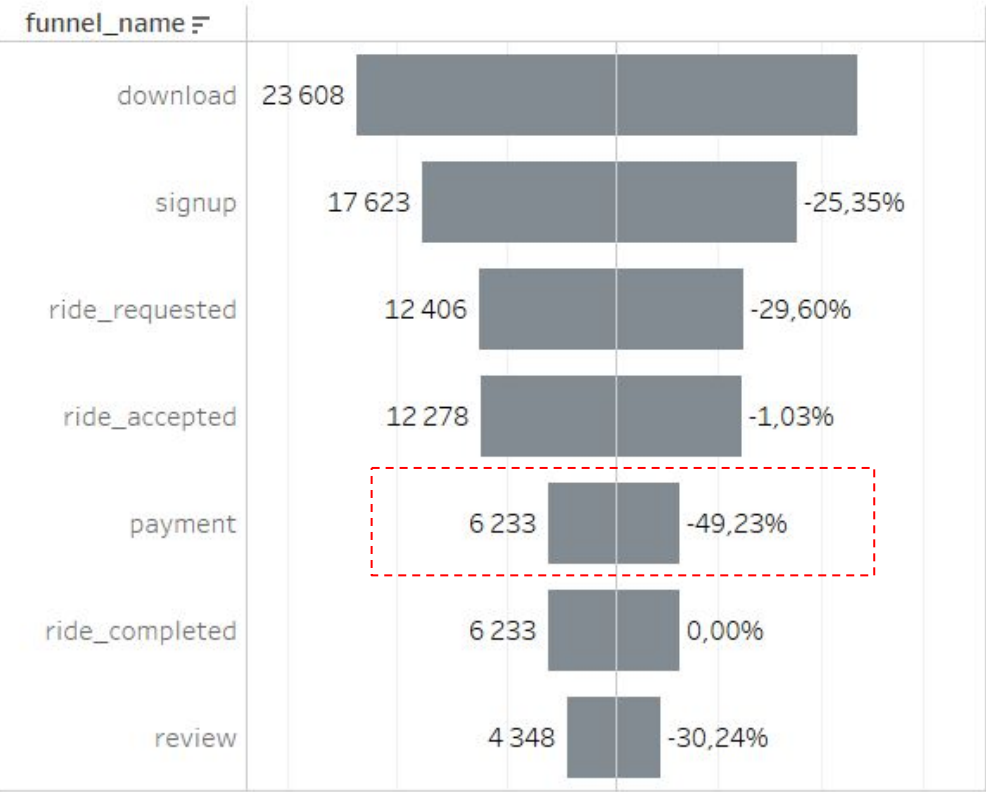
---

## **Вхідні дані:**

Інформація про завантаження додатку, про реєстрацію нових користувачів, про запити поїздки, дані про фінансові операції та відгуки користувачів після поїздки.

**Період:** 01.01.2021 - 31.12.2021

# Воронка користувача



**26,4%**  
(6 233 клієнтів)

пройшли етап від завантаження додатку до завершення поїздки.

**69,8%**  
(4 348 клієнтів)

після завершення поїздки залишили відгук.

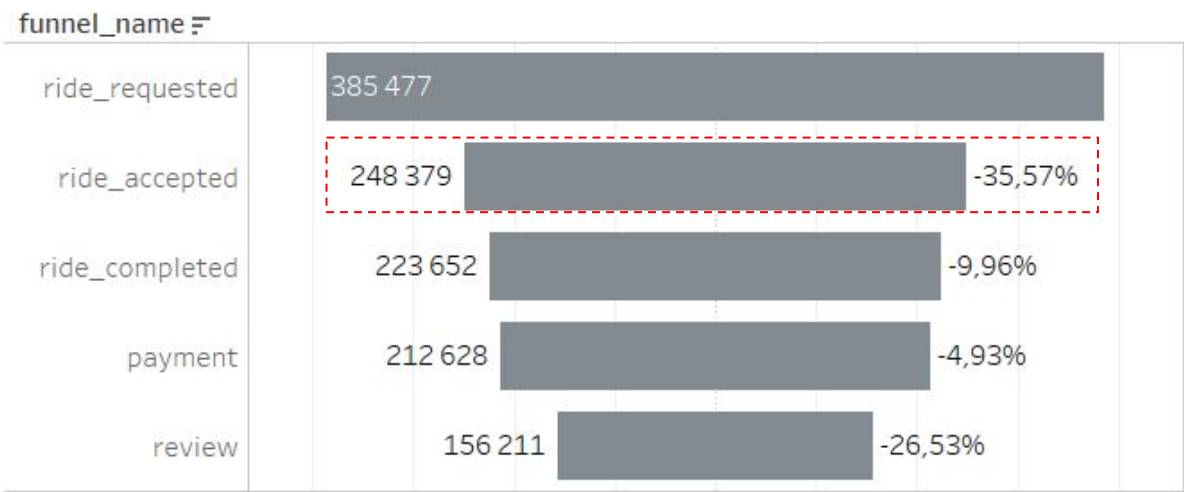
На загальному пулі клієнтів найбільш “просідаючий” етап - оплата поїздки (payment).

# Воронка поїздок

**55,2%**  
(212 628 поїздки) поїздок завершуються успішно

**73,5%**  
(156 211 поїздки) поїздок після завершення залишили відгук

## Funnel of rides

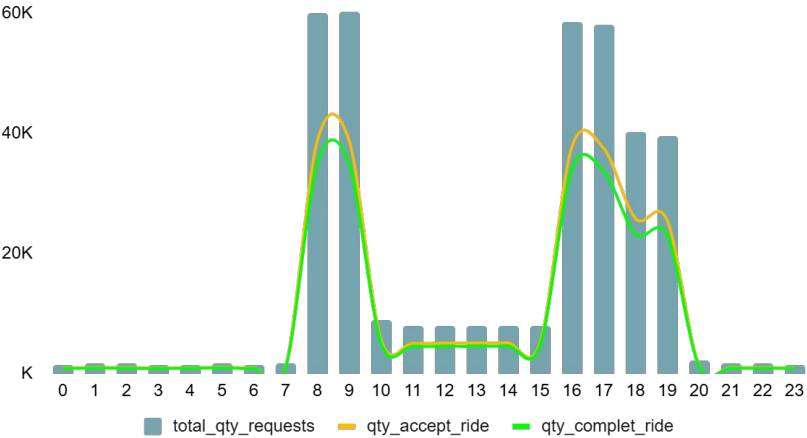


На загальному пулі поїздок найбільш “просідаючий” етап - підтвердження поїздки (ride\_accepted).

# Вузькі місця (drop-off points)

- Які етапи найбільше “просідають”?
- Чи є залежність від часу доби, дня тижня чи очікування водія?

Rides by hours of the day

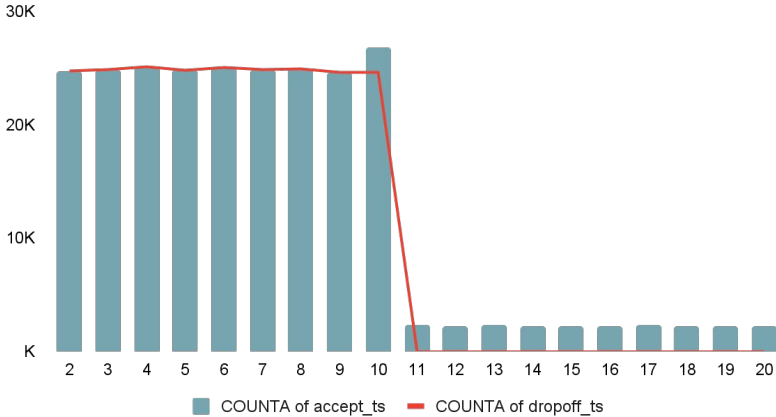


Спостерігаємо пікові навантаження протягом доби в проміжках з 8 до 9 та з 16 до 19 годин.

Проте % відхилення заявок протягом кожного часу доби знаходиться в межах від 41% до 42%, що свідчить про те, що **піковість годин не впливає на відхилення заявок.**

**День тижня також не впливає на кількість заявок чи їх відхилення.**

AVG accept time (minutes)



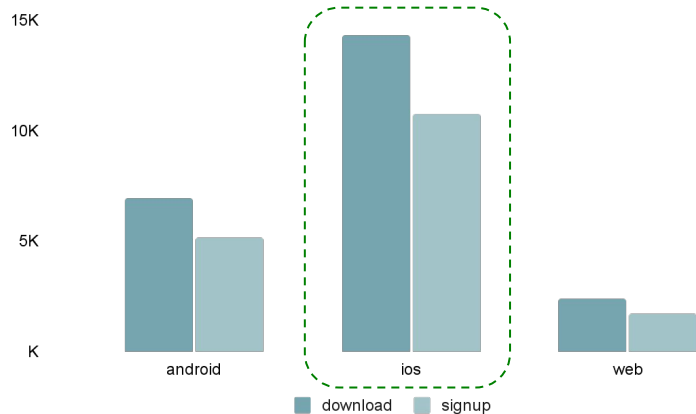
Клієнти готові на 100% чекати підтвердження заявки максимум 9 хвилин.  
Лише 8% клієнтів скасовують поїздку при середньому очікуванні 10 хв.

**Рекомендація:**  
зменшити час підтвердження заявки мінімум до 9 хв.

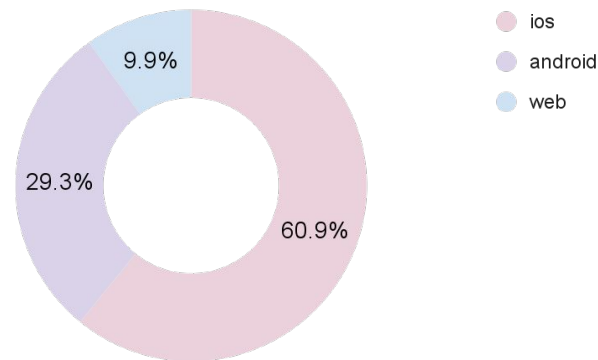
# Аналіз ефективність платформ

На які платформи варто спрямувати маркетинговий бюджет: iOS, Android чи Web?

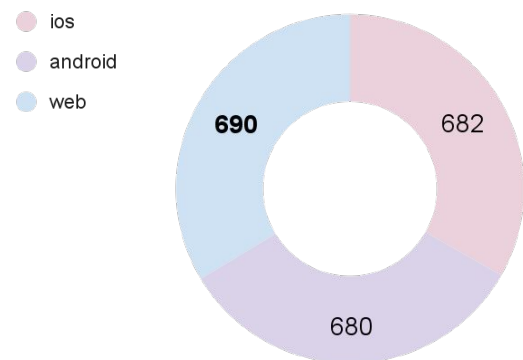
Download and signup by platform



% completed rides by platform



AVR Revenue per User



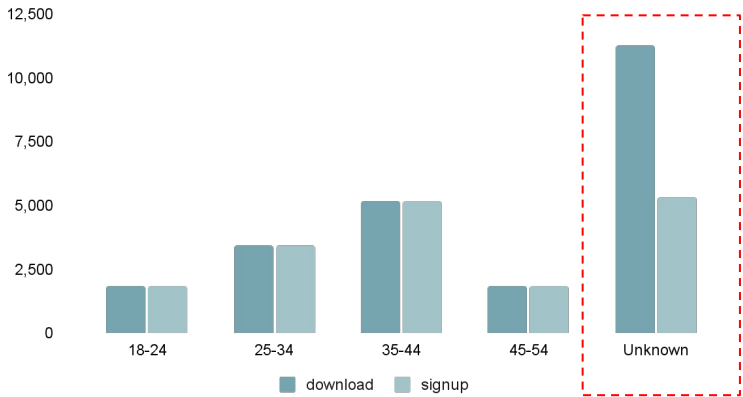
Найбільш популярною є платформа IOS і вона дає найбільшу конверсію при реєстрації (75%) та найбільшу к-ть успішних (завершених) поїздок (61%). Проте найбільший середній дохід з клієнта дає платформа Web.

**Рекомендація:**  
однозначно IOS - платформа куди варто спрямовувати маркетинговий бюджет. Також варто приділити увагу і платформі Web.

# Аналіз вікових груп користувачів

- Які групи найактивніше користуються сервісом?
- Хто найчастіше доходить до завершення поїздки?

Download vs Signup by age



age_range	uniq_user	total_requests	completed_rides	total_requests vs completed, %
18-24	1,300	40,620	24,046	59%
25-34	2,425	75,236	44,121	59%
35-44	3,662	114,209	66,853	59%
45-54	1,285	39,683	22,675	57%
Unknown	3,734	115,729	65,957	57%
Total	12,406	385,477	223,652	

53% клієнти Unknown завантаживши додаток **НЕ** реєструються в ньому.

Клієнти усіх вікових категорій використовують додаток однаково та однаково завершують поїздки. Проте за кількістю клієнтів переважають категорії 35-44 та Unknown.

**Рекомендація:**

- при реєстрації зробити поле “Вікова категорія” обов’язковим для заповнення, щоб чітко розуміти хто наші клієнти.
- за необхідності доповнити (розширити) перелік вікових категорій, якщо 45-54 це остання категорія в списку.

*В подальшому це дозволить зробити більш поглиблений аналіз причин, чому клієнти не реєструються або з якими складнощами вони*

---

# Додатки

---



# Вузькі місця (drop-off points) скрипти

## Для графіку Rides by hours of the day

```
SELECT
    TO_CHAR(request_ts, 'HH24')                as Hour
, case TO_CHAR(request_ts, 'D')
    when '1' then 7 -- неділя
    when '2' then 1 -- понеділок
    when '3' then 2 -- вівторок
    when '4' then 3 -- середа
    when '5' then 4 -- четвер
    when '6' then 5 -- п'ятниця
    when '7' then 6 -- субота
    else null end                             as Day
, count(ride_id)                             as total_QTY_requests
, count(case when accept_ts is not null then ride_id end) as QTY_ACCEPT_RIDE
, count(case when dropoff_ts is not null then ride_id end) as QTY_COMPLET_RIDE
FROM
    ride_requests
GROUP by 1,2;
```

## Для графіку AVG accept time (minutes)

```
SELECT
    TO_CHAR(request_ts, 'HH24')                as Hour
, case TO_CHAR(request_ts, 'D')
    when '1' then 7 -- неділя
    when '2' then 1 -- понеділок
    when '3' then 2 -- вівторок
    when '4' then 3 -- середа
    when '5' then 4 -- четвер
    when '6' then 5 -- п'ятниця
    when '7' then 6 -- субота
    else null end                             as day
, AVG(
    (EXTRACT(SECOND FROM (accept_ts - request_ts)) +
    EXTRACT(MINUTE FROM (accept_ts - request_ts)) * 60 +
    EXTRACT(HOUR FROM (accept_ts - request_ts)) * 3600 +
    EXTRACT(DAY FROM (accept_ts - request_ts)) * 86400) / 60
) AS avg_accept_time_minutes
, ride_id
, accept_ts
, request_ts
, dropoff_ts
FROM
    ride_requests
where accept_ts IS NOT NULL
GROUP by 1,2,4,5;
```

# Аналіз ефективність платформ

скрипти

Для графіку % completed rides by platform

```
SELECT
    t1.platform,
    COUNT(DISTINCT t3.ride_id) AS total_ride_requests,
    COUNT(DISTINCT CASE WHEN t3.pickup_ts IS NOT NULL AND
t3.dropoff_ts IS NOT NULL THEN t3.ride_id END) AS
total_completed_rides
FROM
    app_downloads as t1
JOIN
    signups as t2 ON t1.app_download_key = t2.session_id
JOIN
    ride_requests as t3 ON t2.user_id = t3.user_id
GROUP BY
    t1.platform
ORDER BY
    total_completed_rides DESC;
```

Для графіку AVR Revenue per User

```
SELECT
    t1.platform
, COUNT(DISTINCT t2.user_id) AS uniq_users
, SUM(t4.purchase_amount_usd) AS total_revenue_usd
, SUM(t4.purchase_amount_usd) / COUNT(DISTINCT t2.user_id)
AS avg_revenue
FROM
    app_downloads as t1
JOIN
    signups      as t2 ON t2.session_id = t1.app_download_key
JOIN
    ride_requests as t3 ON t3.user_id = t2.user_id
JOIN
    transactions as t4 ON t4.ride_id = t3.ride_id
where t4.charge_status = 'Approved'
GROUP BY
    t1.platform;
```

# Аналіз вікових груп користувачів

скрипти

Для таблиці

**SELECT**

*t1.age\_range,*

**count** (**distinct** *t1.user\_id*) **as** *Uniq\_USER,*

**count**(*t2.ride\_id*) **as** *total\_requests,*

**count**(**case when** *t2.pickup\_ts* **is not null and** *t2.dropoff\_ts* **is not**  
**null and** *t2.cancel\_ts* **is null then** *t2.ride\_id* **end**) **as**

*completed\_rides*

**from** *signups* **as** *t1*

**join** *ride\_requests* **as** *t2* **on** *t2.user\_id* = *t1.user\_id*

**group by** *t1.age\_range;*