Slide 1

Project 1

Cohort Analysis of Ironhack Payments Users

Ironhack Data Science and Machine Learning Bootcamp

Author: Ginosca Alejandro Dávila

Date: December 13, 2024

speaker's note

Hello everyone, and thank you for being here.

This presentation is part of my first project for the Ironhack Data Science and Machine Learning Bootcamp, and it focuses on a Cohort Analysis of Ironhack Payments Users.

My name is Ginosca Alejandro Dávila, and I’m excited to walk you through the insights and learnings I’ve uncovered during this analysis. Let’s begin.

slide 2

Project Overview

* **Project:** Cohort Analysis of Ironhack Payments Users
* **Goal:** Understand user behavior over time, segmented by their first cash request month
* **About Ironhack Payments:** Ironhack Payments is a digital financial services company offering *short-term cash advances* with *transparent pricing*. Users can:
  + Request *regular (free)* or *instant (fee-based)* transfers
  + Repay via scheduled *SEPA direct debits*
  + Encounter *fees* for instant transfers, postponements, or failed payments

Speaker’s note  
  
This project centers on a **cohort analysis** of Ironhack Payments users. The main objective is to understand how user behavior evolves over time, segmented by the **month of their first cash advance request**.

Ironhack Payments is a **digital financial services platform** that offers short-term cash advances. Their business model emphasizes **transparent pricing** without traditional interest — instead, users may pay **fees** for certain services.

Users have a few key interactions with the platform:

* They can request **regular (free)** or **instant (fee-based)** transfers.
* Repayments are made via **scheduled SEPA direct debits**.
* And they may encounter **fees** if they request instant transfers, postpone repayment dates, or if a payment incident occurs.

Understanding these user flows helps guide the metrics we analyze and provides context for behavioral trends.

Slide 3

**Key Metrics Analyzed**

📈 Frequency of Service Usage

Measure how often users from each cohort use Ironhack Payments’ cash advance services over time.

⚠️ Incident Rate

Track the rate of payment incidents (e.g. failed reimbursements) by cohort to identify trends or risk patterns.

💰 Revenue Generated

Calculate the total fees collected per cohort to assess the financial contribution of each user group.

🧠 New Relevant Metric

Includes an additional user behavior or performance metric tailored to reveal deeper insights beyond standard KPIs.

Speaker’s note

In this project, we’re focusing on four key metrics to evaluate user behavior and cohort performance.

The first is **Frequency of Service Usage** — we measure how often users from each cohort request cash advances over time. This helps us understand engagement and usage longevity.

Next, we calculate the **Incident Rate**, which tracks failed reimbursements or other payment issues by cohort. This reveals patterns of risk and highlights areas for operational improvement.

The third metric is **Revenue Generated**. We sum the total amount of fees paid by each cohort to understand their financial contribution to the business.

Lastly, we include a **Custom Metric**. This will be a tailored indicator that goes beyond standard KPIs — it could focus on behavioral patterns, retention trends, or a unique insight that emerges from the data.

Together, these metrics provide a well-rounded view of user lifecycle, risk, and value.

Slide 4

**Dataset Summary**

🧾 Data Source

* Provided by Ironhacks Payments as raw .csv and .xlsx files
* Definitions and business context available in **Lexique - Data Analyst.xlsx**

📄 Cash Requests (cash\_df):

* 23,970 rows × 16 columns
* Contains one row per cash advance request
* Includes: cash request ID, amount,status, timestamps, user IDs, transfer type, reimbursement info, recovery info

💸 Fees Dataset (fees\_df)

* 21,061 rows × 13 columns
* Contains one row per fee applied to a cash request
* Includes: ,fee ID, cash request ID, fee type, status, reason, category, timestamps, amount and charge timing

Speaker’s notes

Our project is based on two primary datasets provided by Ironhack Payments in CSV format, along with a Lexique file that explains the business context and column definitions.

On the left, we have the cash\_df dataset, which contains nearly 24,000 rows — one per cash advance request. Each row includes the request ID, amount, status, creation and update timestamps, user identifiers, transfer type, reimbursement information, and recovery status in case of payment incidents.

On the right, we have the fees\_df dataset, with just over 21,000 rows. Each row corresponds to a specific fee applied to a cash request. This includes fee ID, type, status, reason, category, and when the fee was charged.

Understanding the structure of these datasets is key for both our data cleaning process and the subsequent cohort analysis.

Slide 5

**Data Quality Assessment**

* ✅ Checked for **missing values**
  + **Mostly expected (e.g.**  reimbursement\_date **missing for requests that were rejected, canceled, or already reimbursed)**
  + **Some required deeper inspection** (e.g., user\_id, cash\_request\_id)
* ✅ Verified **datetime columns**
  + **Inconsistent formats and some timezone-aware vs. naive values**
  + **No future-dated timestamps found beyond 2024-12-12**
* ✅ Validated **monetary fields**
  + **No negative or zero values in** amount or total\_amount
* ✅ Checked **link integrity** between datasets
  + Found 4 fees\_df rows with missing cash\_request\_id
* ✅ Duplicate analysis
  + No fully duplicated rows or duplicate primary keys found

Speaker’s note

In this phase, I focused on understanding the reliability of the raw data before applying any cleaning steps.

I started by reviewing **missing values**. Most of them were expected — for example, the reimbursement\_date is often missing for requests that were either canceled, rejected, or already reimbursed. Others like user\_id and cash\_request\_id required deeper inspection to ensure they wouldn’t break links or logic downstream.

I also validated all **datetime fields**, where I found a mix of timezone-aware and naive formats. After normalizing them, I verified that there were no timestamps mistakenly placed in the future.

For **monetary fields**, I confirmed that all values in amount and total\_amount were valid — no negatives or zeroes.

I checked for **dataset link integrity** and found just four rows in the fees dataset missing a cash\_request\_id. These couldn’t be linked to any cash request and were flagged for removal.

Finally, I performed a **duplicate check**, and I’m happy to report there were no fully duplicated rows or primary key conflicts.

This assessment gave me a clear understanding of the key quality issues to address in the next step: data cleaning.

Slide 6

**Data Cleaning Summary**

* ✅ Converted all **date columns** to be timezone-naive (datetime64[ns])
* ✅ Stripped inconsistent timezones for uniformity
* ✅ Unified user\_id and deleted\_account\_id into final\_user\_id
* ✅ Removed 4 unlinkable rows from fees\_df
* ✅ Standardized **categorical values** (e.g., status, type)
* ✅ Verified and preserved **monetary columns** (amount, total\_amount)
* ✅ Reordered columns for analysis readability

Speaker’s note

Based on the quality checks, I applied several cleaning steps. All datetime fields were converted and normalized for consistent use. I created a unified final\_user\_id to ensure we could group users accurately, regardless of account deletion.

I also removed rows that couldn’t be linked across datasets, and standardized categorical values to prevent mismatches during grouping. Finally, I verified the numeric columns and reordered variables to improve dataset usability going forward.

Slide 7

**Exploratory Data Analysis Overview**

**🔎 EDA Objectives**

* Understand request volumes, fee types, and user behavior
* Uncover time-based trends in requests, revenue, and incidents
* Prepare data for cohort definition and metric aggregation

**📁 Cleaned Datasets Used**

* clean\_cash\_requests.csv: 23,970 rows
* clean\_fees.csv: 21,057 rows

Speaker’s note  
  
Before jumping into cohort analysis, I conducted exploratory data analysis to better understand the structure and behavior of the cleaned datasets. This helped me surface trends in usage, revenue, and platform risks. The data used here was the output of the previous cleaning phase.

Slide 8

**Request and Revenue Growth Over Time**

**📈 Key Insights**

* Strong growth began in May 2020, accelerating through the summer
* October 2020 marks the peak in both requests and cash volume
* Request volume and requested amounts mirror each other closely

📆 Note: Nov 2019 and Nov 2020 are partial months

Time Series Graph: Monthly Cash Requests Volume and Total Requested Amount

Speaker’s note

We can clearly see a strong upward trend starting in May 2020, with both the number of requests and the total cash amounts rising steadily each month. The chart shows how closely these two trends mirror each other — when more users requested cash, they also requested higher total amounts.

The platform reached its peak in October 2020, both in terms of user activity and financial demand. It’s important to note that the first and last months shown — November 2019 and November 2020 — are partial months, which explains their lower numbers.

These patterns give us a solid foundation to define user cohorts by month and better understand behavior over time.

Slide 9

**User Activity and Behavior**

* 11,793 total unique users
* 60.5% used the service only once
* Remaining 39.5% used it multiple times

Bar Graph: Distribution of Cash Requests per User

Speaker note

Most users are one-time users, but nearly 40% returned for additional cash requests. This distribution shows there's a meaningful segment of repeat users, which could impact cohort retention and revenue metrics

Slide 10

**Transfer Preferences Over Time**

**🚀 Instant vs Regular Transfers**

* Instant transfers introduced around July 2020
* Instant share increased to 93% by October 2020
* Regular transfers declined as instant became dominant

Time series graph: Instant vs Regular Transfers and Incident Rate (%)

Speakers note

This shift shows a clear preference for faster service, despite the fee. Instant transfer usage rose from 0% to over 90% in just a few months, influencing both revenue and operational patterns.

Slide 11

**Revenue Breakdown by Fee Type**

* Instant fees: €55,480
* Postpone fees: €38,830
* Incident fees: €10,980

Bar graph: Total Revenue by Fee Type

Speakers note

Instant fees are the largest source of revenue, followed by postpone and incident fees. This confirms that the fee model works primarily through user preference for instant service, rather than penalties or late payments.

Slide 12

**Incidents and Risk Trends**

**⚠️Incident Insights**

* 86% of requests had no incidents
* Peak incident volume in October 2020
* Incident rate remained stable around 10–16%

Stacked bar graph: Monthly Cash Requests: Incident vs Non-Incident

Speaker’s note

While most requests are completed without issue, the number of incidents grew in line with user activity. Still, the incident ratio stayed relatively stable, suggesting a manageable risk profile despite growth.

Slide 13

**Revenue Timing and Fee Consistency**

**📈 Key Insights**

* Instant and incident fees are always charged after request
* Postpone fees can be charged before or after
* 99.99% of fees are flat €5 (only one €10 outlier)

Side by side bar graph: Total Revenue by Fee Type and Charge Moment

Bar graph: Frequency of Fee Amount (5 euros vs 10 euros)

Speaker’s note

Here we see how the timing of fees relates to user behavior — incident and instant fees are charged after the request, while postpone fees are more flexible. It’s also worth noting that nearly all fees are set to a flat €5, showing a simple, predictable pricing structure, which could positively influence user trust.

Slide 14

**Linking Requests to Revenue**

**🔗 Merged Dataset Overview**

* Merged on cash\_request\_id
* 23,970 cash requests, 12,933 had fees (~54%)
* Resulting dataset: 32,094 rows x 33 columns

Speakers note

By merging the datasets, I was able to trace which requests triggered which fees. This linkage is crucial for analyzing user behavior and financial impact at the request level, enabling better cohort-based analysis going forward.

Slide 15

**Ready for Cohort Analysis**

**📆 Cohort Preparation Steps**

* Extracted first request date per user
* Assigned each user a cohort month
* Tracked Monthly Active Users (MAU)

Time series graph: Monthly Active Users

Speaker’s note

By merging the datasets, I was able to trace which requests triggered which fees. This linkage is crucial for analyzing user behavior and financial impact at the request level, enabling better cohort-based analysis going forward.