Team B10

Members: Marten Peljo, Patrick Tiit Raal, Rander Erich Pikkani

Repository link: https://github.com/p4triko/Microtransaction-Prediction-IDS25

Business understanding

Background

Many online games (especially free-to-play) rely on microtransactions as a primary revenue stream. If a studio can predict which players are likely to make a purchase (or are close to converting), it can tailor in-game offers, messaging, and content pacing to improve conversion while avoiding “spammy” promotions to players who will not buy. In this project, the goal is to use player/session statistics from the Kaggle dataset to predict whether a player makes microtransaction purchases (binary target).

Business goals

From the perspective of a game studio’s product analytics team:

1. **Targeted monetization:** Predict purchase likelihood so that promotions can be shown to the right players at the right time.

2. **Player experience protection:** Reduce unnecessary or poorly timed monetization prompts for unlikely buyers (protecting engagement and sentiment).

3. **Insight for design decisions:** Identify which behavioral patterns correlate with purchases (e.g., progression speed, cadence of sessions) to inform tuning and content strategy, consistent with CRISP-DM’s emphasis on clearly stating the business impact before modeling.

Business success criteria

We will consider the project successful if the model helps the game team **earn more from microtransactions without annoying players**.

If the model were used in the real game, success would mean:

- **More buyers from targeting:** When we show offers/promotions to the players the model says are most likely to buy, **a larger share of them actually buy** compared to showing offers randomly.

- **More revenue per campaign:** The targeted group generates **more microtransaction revenue** (and the extra revenue is worth any discounts or rewards we give).

- **No damage to the game experience:** Targeting should **not reduce normal play**, meaning players should not play less, quit more, or leave worse feedback because of the promotions.

- **Works consistently:** Results should be similar over time (not just “lucky” in one test week).

Since we cannot run real in-game tests, we will treat it as successful if:

- The model can **reliably rank players** so that the “top likely buyers” group contains **far more buyers than average**.
- The model’s reasons are understandable (we can explain which player behaviors are linked to buying).

Assessing your situation

Inventory of resources

- **Data:** Kaggle dataset containing player demographics and gameplay/session statistics plus the microtransaction purchase target.

- **People:** Team B10 (data miners); “stakeholder” role can be assumed by course staff.

- **Tools:** Python + notebooks; standard ML stack; version control (Git).

- **Process guidance:** CRISP-DM structure.

Requirements, assumptions, and constraints

Requirements

- A working, repeatable ML pipeline (clean data → train model → test results).
- Clear evaluation metrics and results on unseen test data.
- A short explanation of what features matter and what the model is doing.

Assumptions

- The dataset correctly shows whether a player bought microtransactions or not.
- The input statistics (playtime, sessions, level, etc.) are available before we make the prediction.
- The data is “close enough” to real players that patterns we learn are meaningful.

Constraints

- This is a public dataset, so we don’t have business details like pricing, promotions shown, or marketing history.
- Purchasers may be a small group, which makes prediction harder.
- We cannot truly prove business impact without running real in-game experiments.

Risks and contingencies

Main risks

- **Too few buyers (imbalanced data):** The model might predict “no purchase” most of the time and still look good.

- **Not transferable:** The model might work only for this dataset, not for another game.
- **Unfair targeting:** The model might behave differently for different demographic groups.

What we do about them

- Use the right metrics for imbalance.
- Be clear that this is a prototype and needs validation on real game data.
- Check performance across groups and consider excluding sensitive features.

Terminology

- **Microtransaction purchaser:** A player who buys something with real money (Yes/No).
- **Precision:** Out of the players we target, how many actually buy?
- **Recall:** Out of all buyers, how many did we successfully find?

Costs and benefits

Costs

- Time to clean data, train models, and write the report.
- Risk of wrong decisions if the model is misleading.
- If used in real life: cost of discounts/rewards and time to run tests.

Benefits

- More efficient offers (show deals to players likely to buy).
- Higher conversion and revenue compared to random targeting.
- Better understanding of what player behaviors are linked to purchases.
- Fewer annoying promotions shown to players who won't buy.

Data mining goals

- To develop a classification model that predicts whether a player will make a microtransaction purchase.
- Produce probability scores so players can be ordered from most likely to least likely to buy.
- Determine which variables contribute most to predicting purchases.
- Handle class imbalance, ensure proper evaluation and prevent overfitting so the results generalize beyond the training data

Data-mining success criteria

The data-mining portion is considered successful if the model demonstrates strong predictive performance.

- The top-ranked segment (e.g top 10% by predicted probability) should contain a significantly higher proportion of buyers.
- Understandable reasons for predictions, must be able to interpret which behaviours correlate with purchases.
- Running the model again with the same data should produce similar results.
- The model should produce probability scores that could realistically be used for targeted offers in a real game environment.

Data understanding

Gathering the data

1. Outline data requirements

To assess the likelihood of a microtransaction purchase happening, the project requires data describing the following attributes:

- General demographics: Gender, Age, Location
- Classification about the game: Game genre and difficulty
- Behavioral metrics:
    - Playtime, preferred in hours.
    - Sessions, per day / week / month.
    - Session durations, in minutes.
    - Player level, indicating the investment put in.
    - Number of trophies / achievements unlocked.
- Engagement indicators: Level of engagement reached during playtime.
- Target: In-game purchases, a binary value (0 or 1).

These fields are required, because spending money on microtransactions is strongly influenced by metrics like player engagement, progression and the general frequency of play sessions.

2. Verify data availability

The dataset used for this project is a Kaggle player-behavior datase. All necessary data is included in the dataset. License for this dataset is “Attribution 4.0 International”. According to the license, the data is free to use and can be built upon, but requires: to give credit, link the license and an indication if changes were made.

3. Define selection criteria

- The target variable is the column indicating In-game purchases, (0 - hasn’t purchased, 1 - has purchased).
- Columns that offer no predictive value don’t need to be included, for example a column showing the player’s ID.

- All rows will be included initially when doing predictions, if something faulty appears, it can be later removed.

Describing the data

The dataset contains 40034 rows and 13 columns. Below are the initial summaries of the features of the given dataset.

Feature	**Data type**	**Description**
PlayerID	Integer	Unique identifier for each player.
Age	Integer	Age of the player.
Gender	Categorical	Gender of the player.
Location	Categorical	Geographic location of the player.
GameGenre	Categorical	Genre of the game the player is engaged in.
PlayTimeHours	Float	Average hours spent playing per session.
InGamePurchases	Binary / Boolean (0 or 1)	Indicates whether the player makes in-game purchases.
GameDifficulty	Categorical	Difficulty level of the game.
SessionsPerWeek	Integer	Number of gaming sessions per week
AvgSessionsDurationMinutes	Integer	Average duration of each gaming session in minutes.
PlayerLevel	Integer	Current level of the player in the game.
AchievementsUnlocked	Integer	Number of achievements unlocked by the player.
EngagementLevel (target)	Categorical	Categorized engagement level reflecting player retention ('High', 'Medium', 'Low')

Exploring Data

Early observations:

- Age seems to range from mid-teens to the late 40s.
- Player commitment is quite varied, deduced from Features PlayTimeHours, SessionsPerWeek and AvgSessionDurationMinutes.
- Player level is also quite varied, but the overall average is more in the mid range.

Exact percentages and stats need to be computed during analysis, but some early deductions could be made from downloaded dataset and analyzing it in Excel.

Exploring the categorical features:

- Location is represented very generally, with 4 values (USA, Europe, Asia and other).
- Gender seems to be distributed quite evenly, but the exact values should be computed at runtime.
- The genre covers many different types of games, but its exact effect can't be deduced at first glance.
- Game difficulty covers 3 levels (easy, medium and hard). Yet again, its impact needs to be extracted later on.
- Engagement level offers similar presentation, as in low, medium and high.

These features may have a high impact during prediction calculation.

Verify data quality

During initial analysis, no missing values seem to exist, but due to the large size of the dataset, it needs to be confirmed with the help of the computer.

However in terms of data presented, some outliers seem to exist. Some players inherit a very high PlayTimeHours value, one appearing very early in the set, PlayerID 9009 having a playtime of 23.94 hours. PlayTimeHours represents the average hours spent playing per session, so that means this particular player is engaged all day every day which is basically impossible. Also looking at the SessionsPerWeek and AvgSessionDurationMinutes, the math

doesn't quite add up for some of these players. These definitely need to be tackled in the data preparation phase.

There is also this incorrect correlation between SessionsPerWeek and the other attributes representing playtime. For some players, for example, let’s take the player with the ID 9013, SessionsPerWeek is assigned to 0, but the AvgSessionDurationMinutes has quite a high value, which doesn’t make much logical sense.

In terms of data preparation and data types, before modeling categorical features need encoding, possibly to an ordinal type. PlayerID can be safely removed, it doesn’t correlate to anything really and it seems to be catered towards just identifying the player, which is not what we're after. Numeric features such as PlayTimeHours and AvgSessionDurationMinutes need handling, possibly converting minutes to hours for easier representation.

Planning your project

Project plan with a list of tasks			
Task	Member 1: Marten	Member 2: Patrick	Member 3: Rander
Choosing the topic	2h	2h	2h
General preparation - creating a repository and a notebook file, syncing everything.	1.5h	1.5h	1.5h
Data analysis	3h	3h	3h
Data preparation	4h	4h	4h
Writing the initial model	8h	8h	8h
Testing the first version of the model	1h	1h	1h
Visualizing the model outputs	1.5h	1.5	1.5h
Adjusting the model based on the initial output	3h	3h	3h
Final tweaks, along with commenting on the workflow for how the model creation was done. Creating a README file for instructions.	2.5h	2.5h	2.5h
Poster design	2h	2h	2h
Presentation	2h	2h	2h
Combined hours:	30.5h	30.5	30.5

List of methods and tools

Tools:

- Google colab, provides a better workflow for a group based project.
- GitHub, project storing and documentation.
- Practice and lecture materials from the course.
- Kaggle documentation about the dataset we chose.

Methods:

- Descriptive statistics for data understanding and data preparations, weeding out improper values/columns. Use basic knowledge, for example means, medians, ranges and primitive correlation logic.
- Modelling for predictions. Random Forest as main prediction model, for learning patterns and probabilities regarding a persons microtransaction purchase likelihood.
- WIth help of confusion matrix, precision, recall, permutations and with ROC curve, do error analysis.
- Plots for displaying results and correlations.