Making the World Playful

How AI automation increases the content relevancy in Candy Crush Saga

King

- Founded in 2003
- 200+ published games, most known – Candy Crush Saga, Candy Crush Soda Saga and Farm Heroes Saga
- Part of Activision-Blizzard since February 2016
- Part of Microsoft since October 2023
- ~ 2000 employees currently

- **Candy Crush Saga** (released 2012)
    - free-to-play tile-matching game
    - >17 000 levels
    - in Europe, teams spread across Stockholm, Barcelona, London, Berlin, Malmö (check out new job openings!)

Candy Crush Saga

Candy Crush Soda Saga

Farm Heroes Saga

Making the World Playful

Making the World Playful

Candy Crush Saga

- Tile-matching game “Match 3”
- Players match candies in combinations of three or more to progress through levels, each with unique challenges and objectives
- Over 17 000 levels, constantly adding new levels to the game

Why we want to integrate AI to Candy Crush and other games?

To enhance experience of players and make the game more relevant for everyone:

Understanding of player preference

Displaying more relevant content

Increase player’s engagement, retention and social interactions

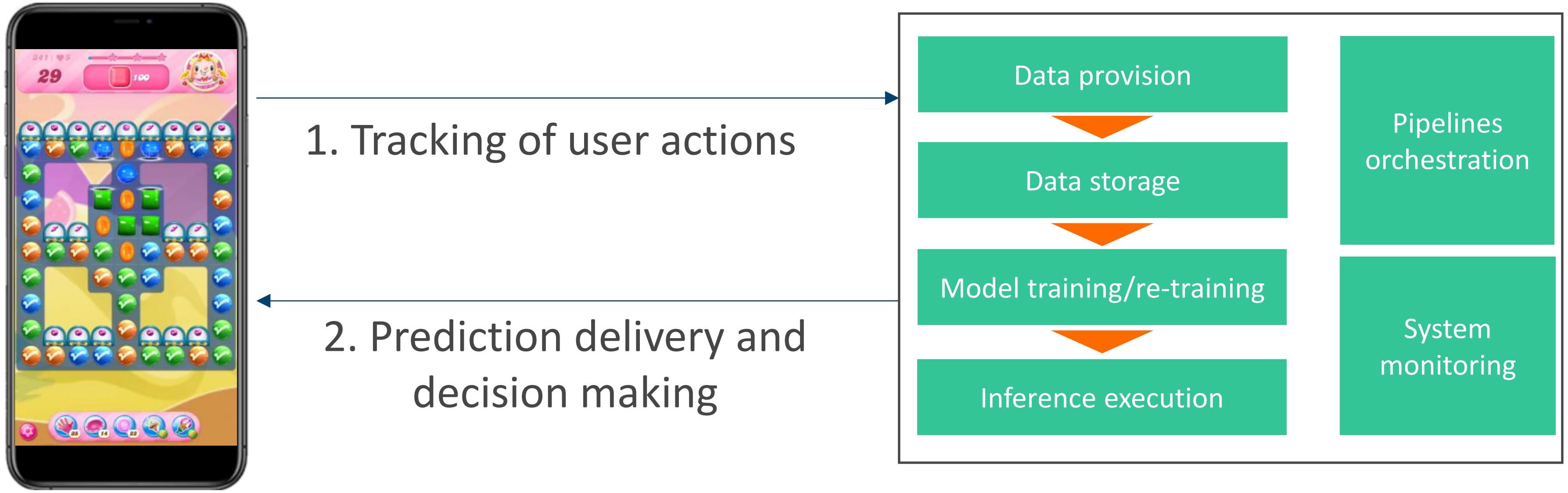
Problem statement – integration of AI solutions is challenging

1. Getting the right data via **tracking**

- There is high data maturity for analytical puposes, but not always for AI/ML pipelines (data granularity, consistency, contextual information)

2. Choosing the right way of **integration**

- Game client built 12 years ago without thinking too much about AI automation (one has to take existing codebase to take into account)



1. Getting the right data via tracking

2. Choosing the right way of integration

3. Use-case example

4. Main takeaways

1. Getting the right data via tracking

2. Choosing the right way of integration

3. Use-case example

4. Main takeaways

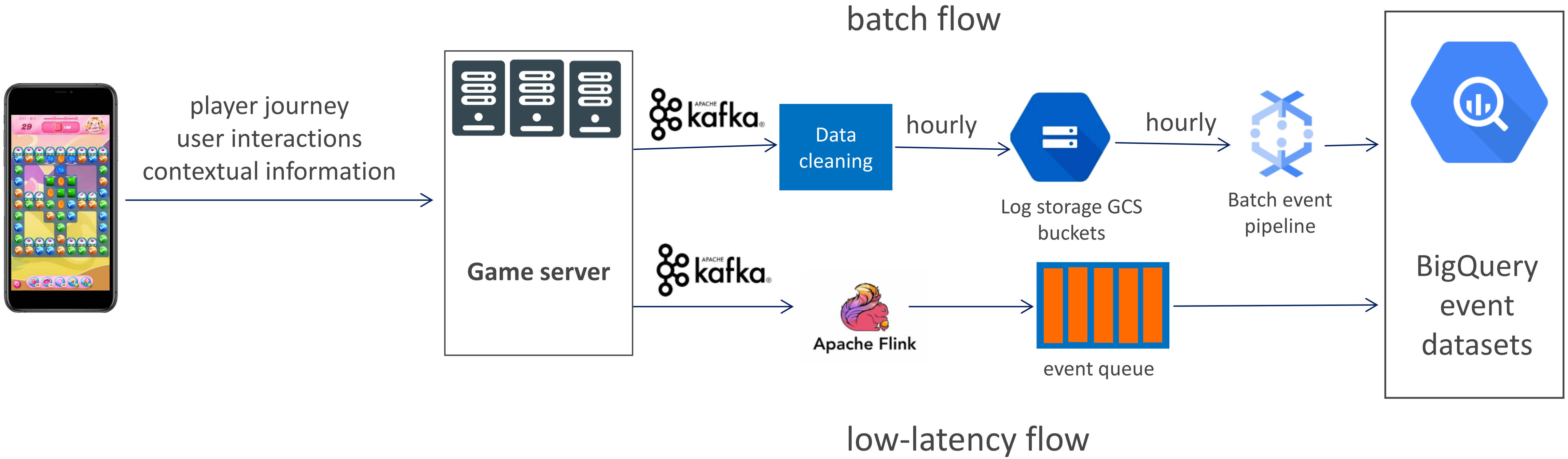
Getting the right data via tracking

- **Challenges**
  - high data maturity for analytical puposes, but not always for AI/ML pipelines (data granularity, consistency – format for different game features, contextual information)
  - data availability – option for both batch and low-latency processing
  - client-server data delivery delay
  - risk of data loss (e.g. due to poor connection of the client)

- **Previously**:
  - every game feature tracked differently

- **Now**:
  - unified template interface and generalizable structures for tracking

Data flow between the client and data warehouse



1. Getting the right data via tracking

2. Choosing the right way of integration

3. Use-case example

4. Main takeaways

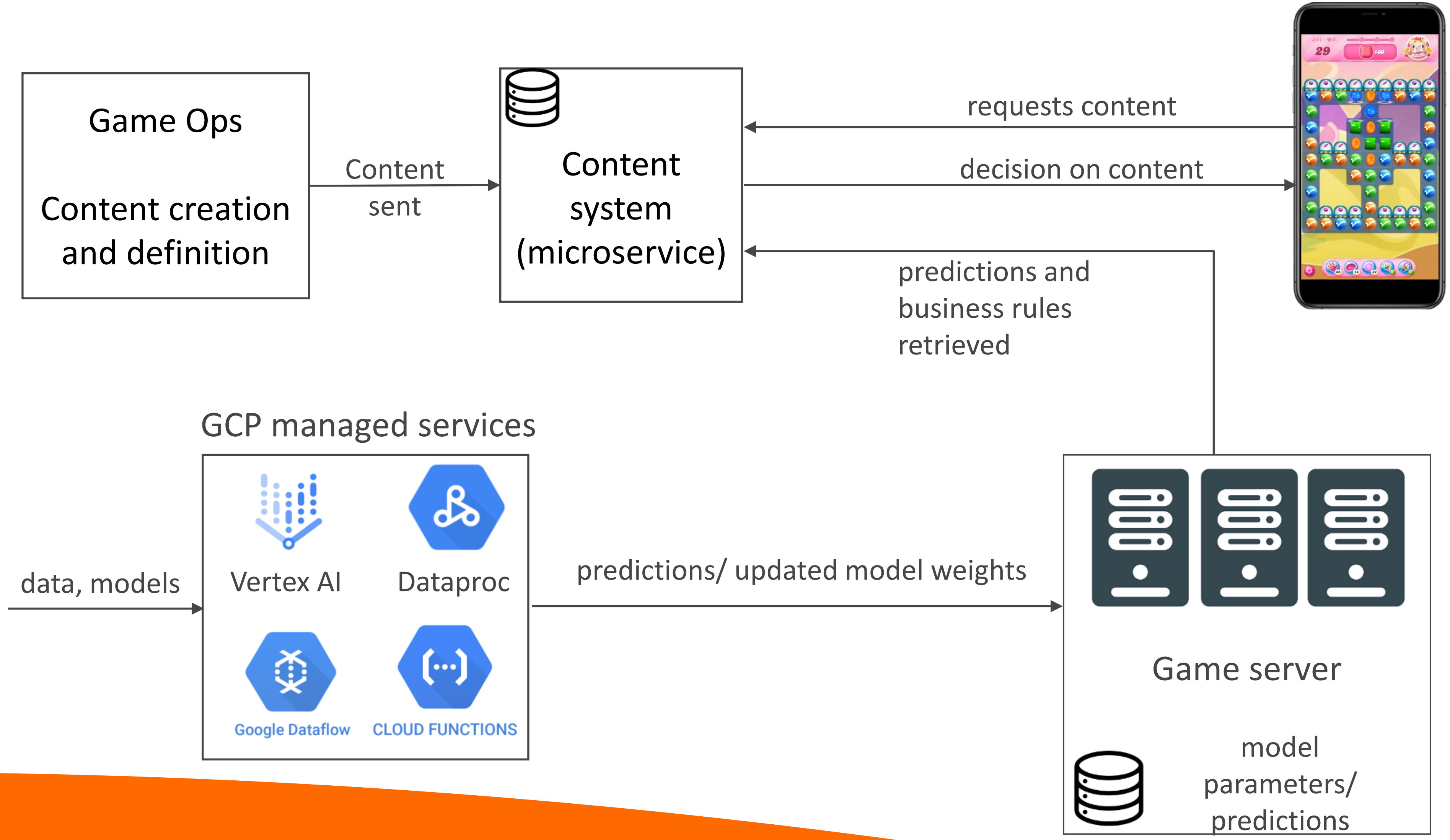
Choosing the right way of integration

- **Challenges**:
  - Integration between Client <-> Game server <-> Data infra (GCP)
  - Predictions freshness (should be always updated, check that they're up to date)
  - User experience (not to let the user wait – e.g. when to retrieve? How to cache?)
  - Fallback (default) option in case prediction pipeline fails or data cannot be delivered

- **Previously**:
  - User lists for campaigns (boolean variables of user belonging to campaign, event …)

- **Now**:
  - Delivery of the tuples (user, prediction) to game server that the game client can regularly retrieve

Delivery of predictions from GCP to the game client



1. Getting the right data via tracking

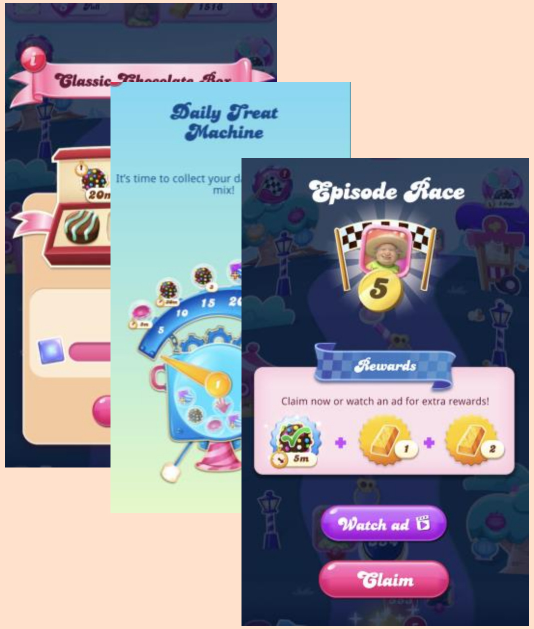
2. Choosing the right way of integration

3. Use-case example

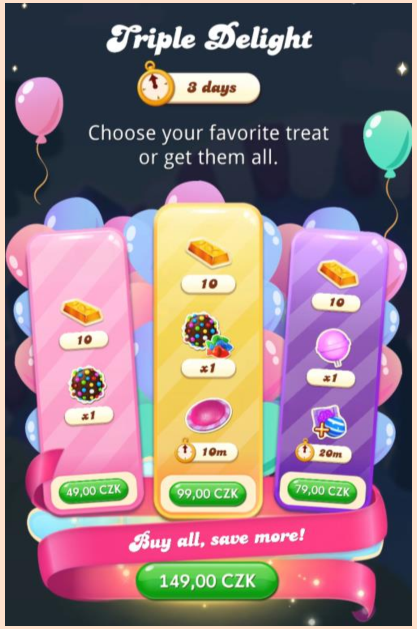
4. Main takeaways

King Making the World Playful

Use–cases: examples



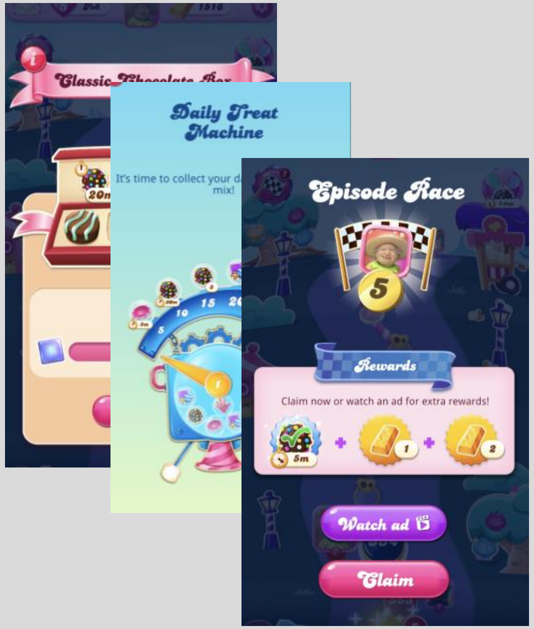
Optimising which pop-ups to show



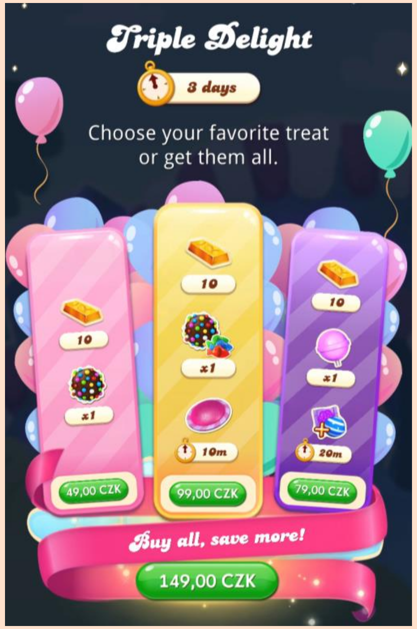
Understanding the user preference for rewards and offers

Displaying the right creative assets

Use–cases: examples



Optimising which pop-ups to show

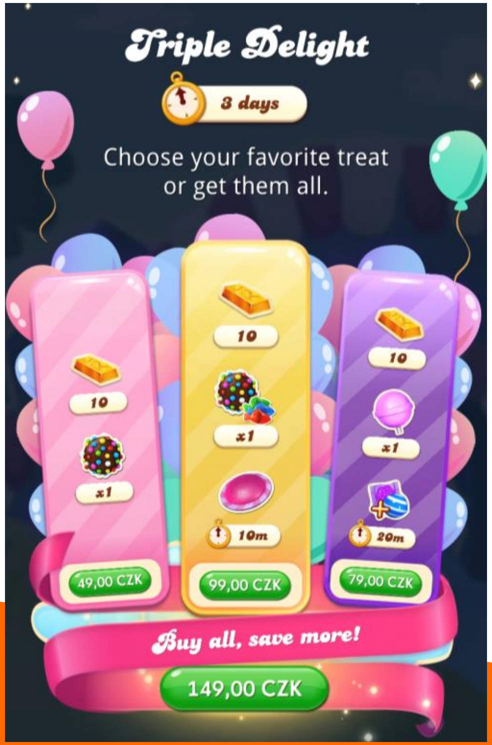


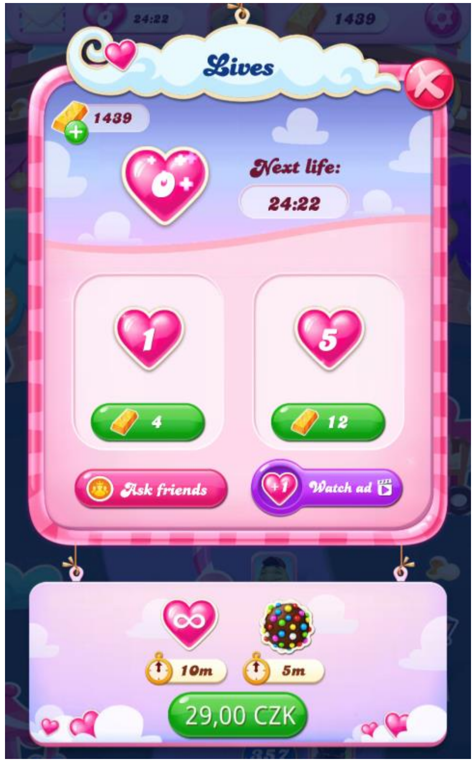
Understanding the user preference for rewards and offers

Displaying the right creative assets

Understanding user preference: Modelling offers and rewards

- users are exposed to many popups with generic contents, especially regarding in-app purchases and rewards
- to keep the game more enjoyable and relevant, it is necessary to display relevant content





Understanding user preference: The problem space is dense




A player can use one or multiple boosters per gameround. This creates a lot of possible combinations.

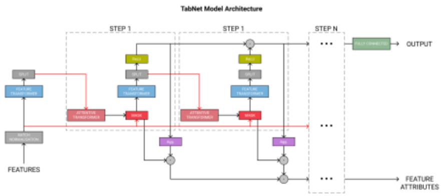




Understanding user preference: Modelling via multivariate regression

Historical activity data

TabNet multi-output regression



Cosine similarity as objective function

$$\frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

for prediction y_i and true label x_i.

$\vec{x}^{PRED}$

Arik, S. O., & Pfister, T. (2019). **TabNet: Attentive Interpretable Tabular Learning**. arXiv. https://doi.org/10.48550/ARXIV.1908.07442

Katsarou, S., Carminati, F., Dlask, M., et al. (2024). **On a Scale-Invariant Approach to Bundle Recommendations in Candy Crush Saga**. arXiv. https://doi.org/10.48550/ARXIV.2408.06799

Tracking and integration implementation

Tracking

- Logging events about user behavior, especially interactions in the game (e.g. GUI tracking)
- Providing contextual information inside the tracking is essential for modelling
- Including client and server timestamps for monitoring any data delays or losses

Integration

- Combination of business rules and predictions into a single pipeline
- Fallback in case the prediction fails or not delivered
- Integrated scalability to be able to consume new predictions (as this is multivariate regression problem)

1. Getting the right data via tracking

2. Choosing the right way of integration

3. Use-case example

4. Main takeaways

Making the World Playful

Main takeaways

Tracking system design should come before developing AI solutions

Integrating predictions for decision making in existing systems requires robust implementation

Latency as performance metric is essential in systems with many connected components for delivering predictions in time

Caching previous predictions and fallbacks are necessary to prevent missing content

Thank you!

Want to discuss tech, explore more ideas or share insights?

Let’s continue the conversation!