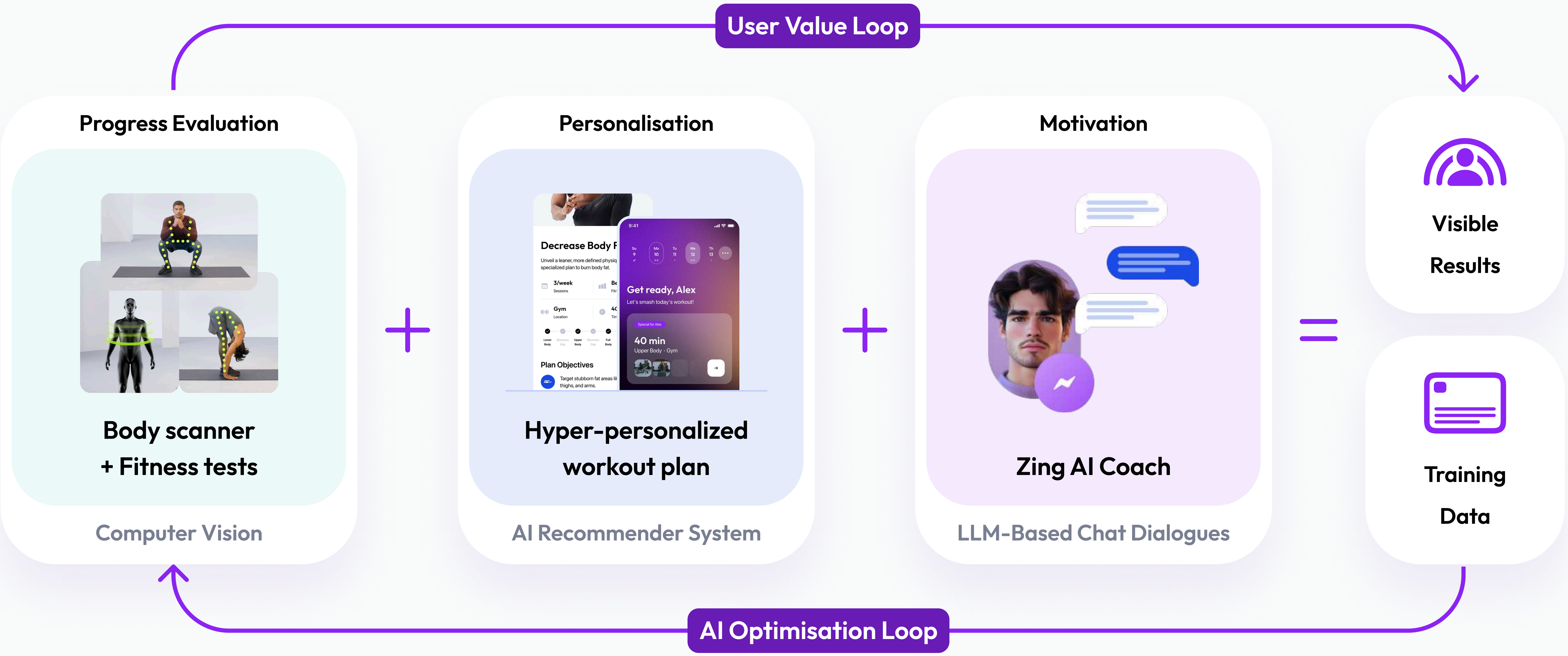
Using Data and AI To Improve Your Fitness.

January 2025

The AI Coach That Works Like a Real Trainer

Everything a real coach offers—personalized guidance, real-time feedback, and motivation—powered by AI



Unique AI Training Data

This unique dataset enables personalized and optimized fitness recommendations, providing a strong competitive advantage

Data points:
2.7m Zing workouts
32m external workouts

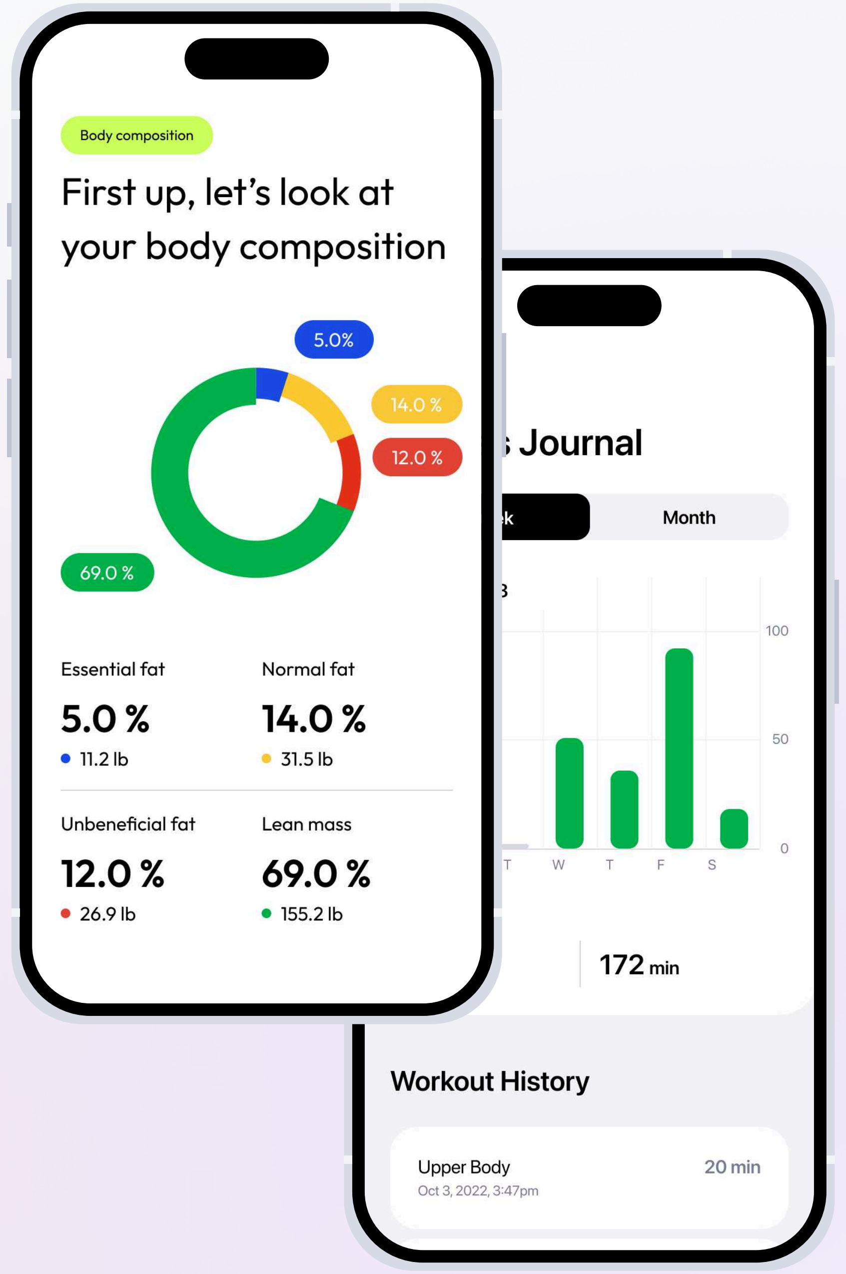
300k body scans
250k fitness tests

Over **30K** monthly fitness dialogues

Data is linked to users' goal progress

Long-term health metrics like sleep and VO2Max for **300K users**

User Data: user profile (age, sex, goal, etc), assessment results (fitness tests, body composition), workout history (logged weights and reps, feedback, HR)



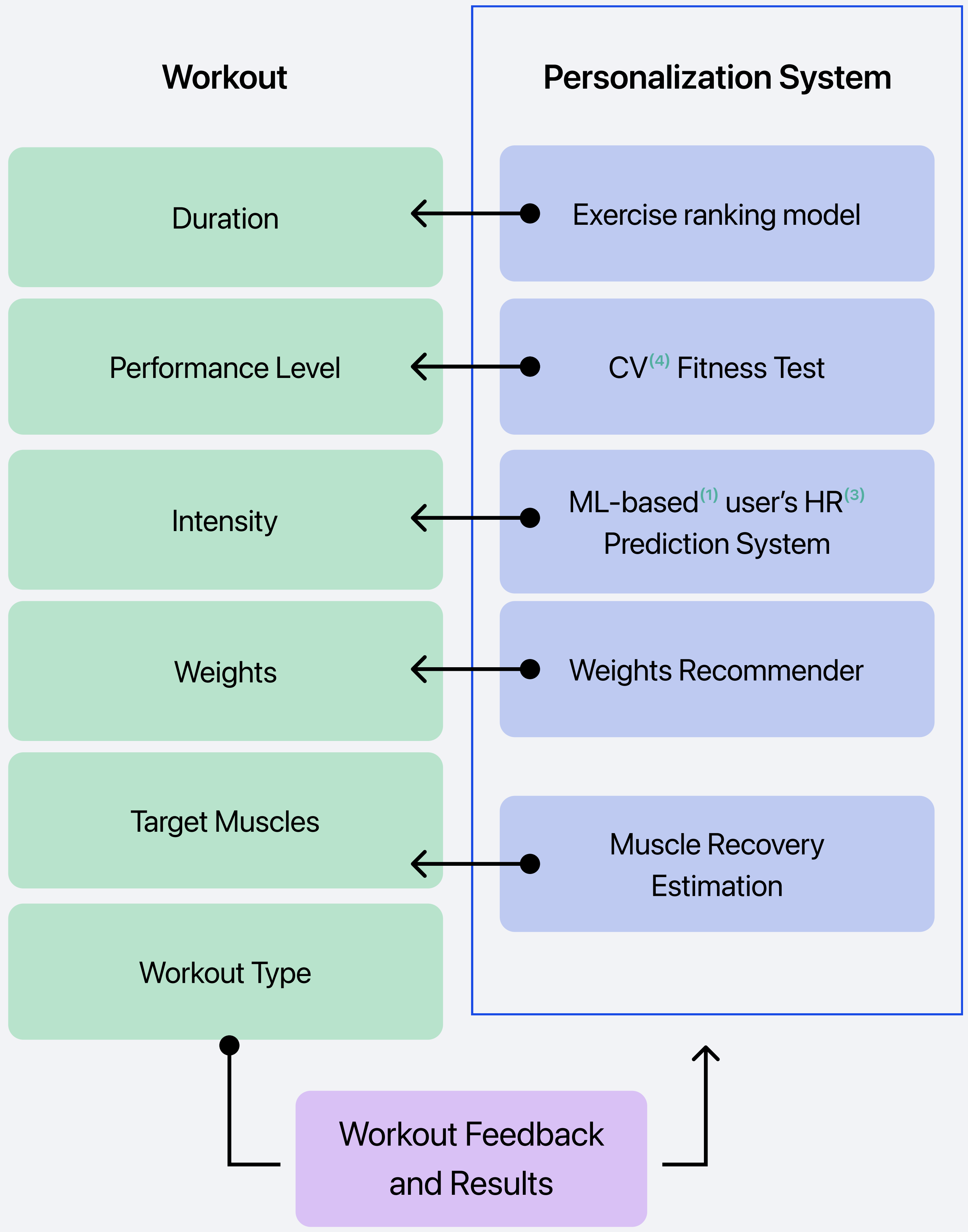
Notes: (1) Share of Monthly Active Subscribers, July 24

How Recommender System Works

Zing advanced ML[1] model constantly optimizes users' workouts based on ability and performance. Zing AI[2] makes training decisions and creates programs at a professional human coach level.

- **Exercise ranking model** is used to suggest best exercises for user based on the profile data.
- **CV-based[3] Fitness Tests** mimic a professional personal trainer's assessment of a person's fitness level.
- **Muscle recovery estimation** optimizes workout timing and target areas for maximum effectiveness and safety.
- **Workout intensity estimation** and users' HR[4] predictions help pick the perfect intensity level that aligns with users' abilities and goals.
- **Weights recommender** suggests initial weight values for each exercise and adjusts them as users progress.

Notes: (1) ML – machine learning. (2) AI – artificial intelligence. (3) CV – computer vision. (4) HR – heart rate.



How intensive is this workout?

1. Dumbbell Goblet Squats – 12-15 reps (Use 10-12 kg dumbbell)
2. Push-Ups – 12-15 reps (Bodyweight or elevate hands for modification)
3. Dumbbell Deadlifts – 12 reps (Use 12-15 kg dumbbells per hand)
4. Bent-Over Dumbbell Rows – 12 reps per side (Use 10-12 kg dumbbells)
5. Plank with Shoulder Taps – 20 taps (10 per side, bodyweight)

Which one is more intensive?

1. Dumbbell Goblet Squats – 12-15 reps (Use 10-12 kg dumbbell)
2. Push-Ups – 12-15 reps (Bodyweight or elevate hands for modification)
3. Dumbbell Deadlifts – 12 reps (Use 12-15 kg dumbbells per hand)
4. Bent-Over Dumbbell Rows – 12 reps per side (Use 10-12 kg dumbbells)
5. Plank with Shoulder Taps – 20 taps (10 per side, bodyweight)

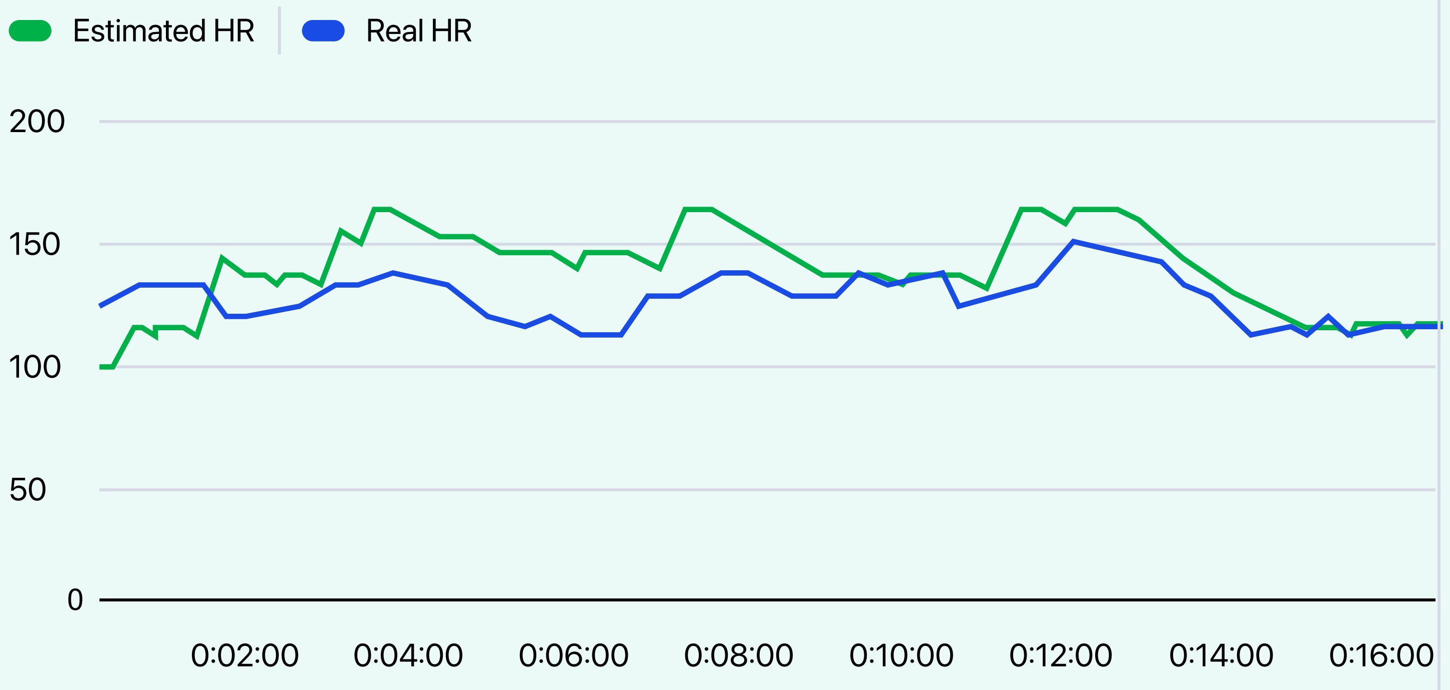
1. Incline Dumbbell Chest Press – 12 reps (Use 12-15 kg dumbbells per hand)
2. Dumbbell Front Raises – 10 reps per arm (Use 6-8 kg dumbbells)
3. Weighted Russian Twists – 20 twists (Use 6-8 kg plate or dumbbell)
4. Plank to Push-Up – 10 reps (Bodyweight)
5. Lunges with Dumbbell Hold – 10 reps per leg (Use 10-12 kg dumbbells per hand)

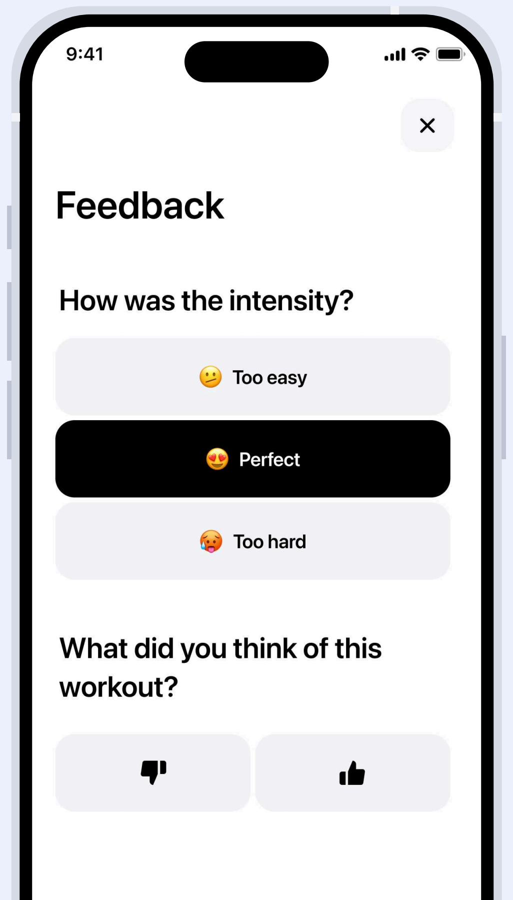
HR-Prediction System and Intensity Recommender

Why it matters: We realized that expert opinions alone couldn't define workout intensity for every user. Each person experiences intensity differently. So, we developed an intensity prediction system that asks users to rate their workout (low, medium, high) after completing it. Expert agreement on intensity was only 66%, showing the importance of personalized feedback.

Takeaway: We estimate the intensity of each exercise and workout using the science-based metric MET[1]. With MET, we can compare workouts to each other and adjust workout intensity by 5% step. Based on the user's profile, we predict HR[2] and cardio zones during each workout with an average error of +/-0.7 zones.

Estimated HR[2] and Real HR[2]





Notes: (1) MET – metabolic equivalent of task ; (2) HR – heart rate.

Sources: (1) Compendium of Physical Activities

How we works with users’ feedback

1. Implicit Feedback Analysis

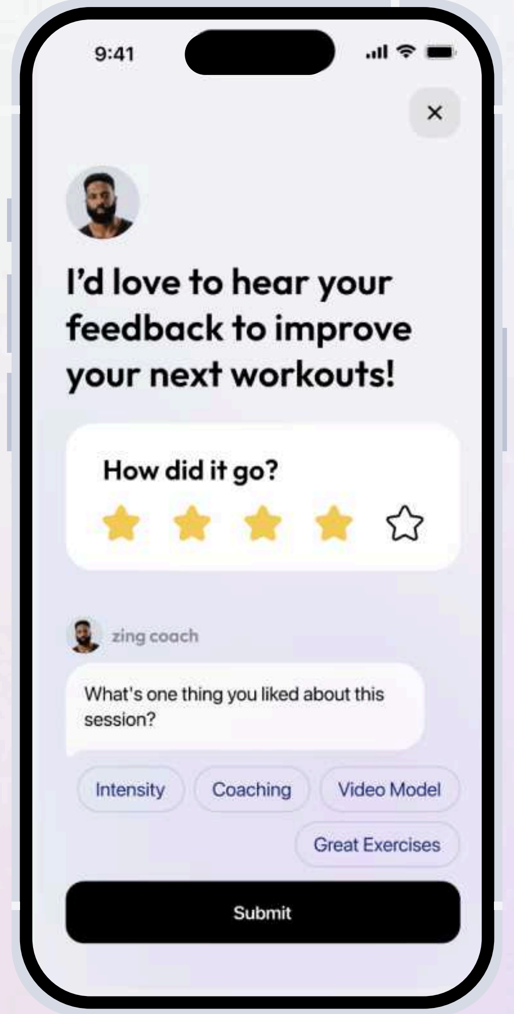
- We process implicit feedback, such as workout patterns and user behavior, using Large Language Models (LLMs).
- LLMs generate summaries that help categorize user sentiments and identify trends effortlessly.

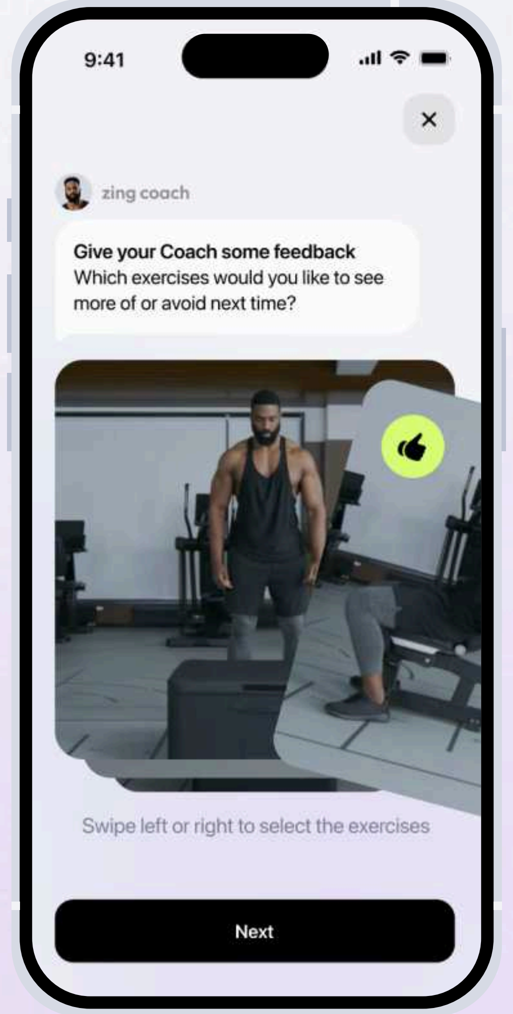
2. In-App Feedback Screens

- After Weight Adjustments: Feedback on weight changes helps refine personalization.
- Post-Workout Feedback: Users rate workouts and intensity, enabling us to tailor future sessions.

3. Exercise-Specific Feedback

- Introducing a new feature: Like/Dislike for Exercises, allowing users to share preferences about individual exercises.
- This helps us personalize routines and improve exercise recommendations for better engagement.






```
Constraints & Scheduling': {'summary': 'A significant number of users report having to end
                                      'their workouts early due to time limitations. This
                                      'suggests a need for more flexible workout durations
                                      'that can be adapted to fit into a busy schedule or
                                      'unexpected interruptions.',
                            'quotes': ['Ran out of time', 'Out of time', 'No time',
                                      'Had to leave', 'Time constraints', 'Gym closed',
                                      'Don’t have time', 'Busy', 'Only had so much time.',
                                      'Had to go take care of something']},
```

Engagement States

Engagement States Research

Objective:

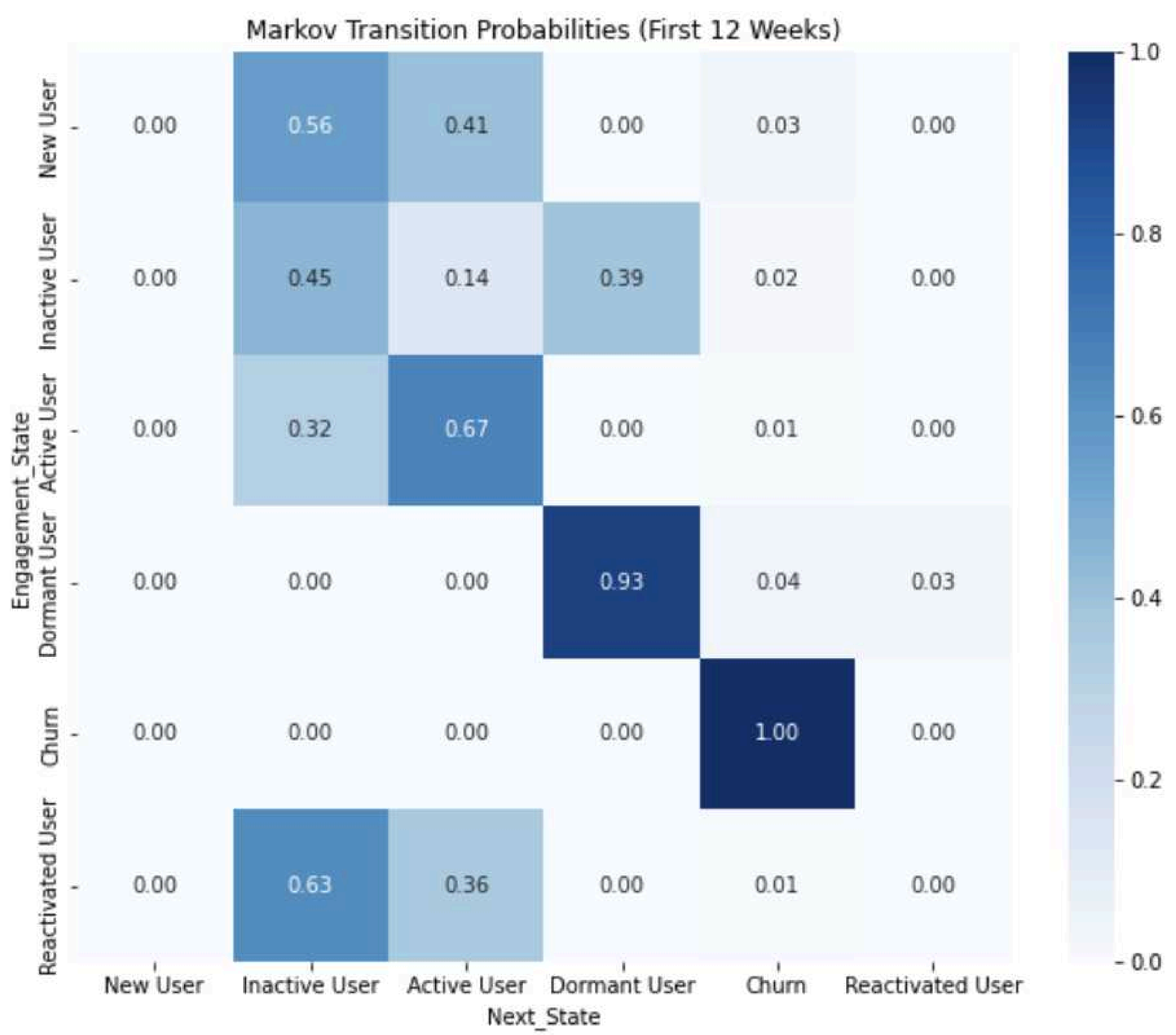
Understand how users move between engagement states over time and use these insights to improve retention. Focus on workout completions as they are the strongest predictor of subscription retention.

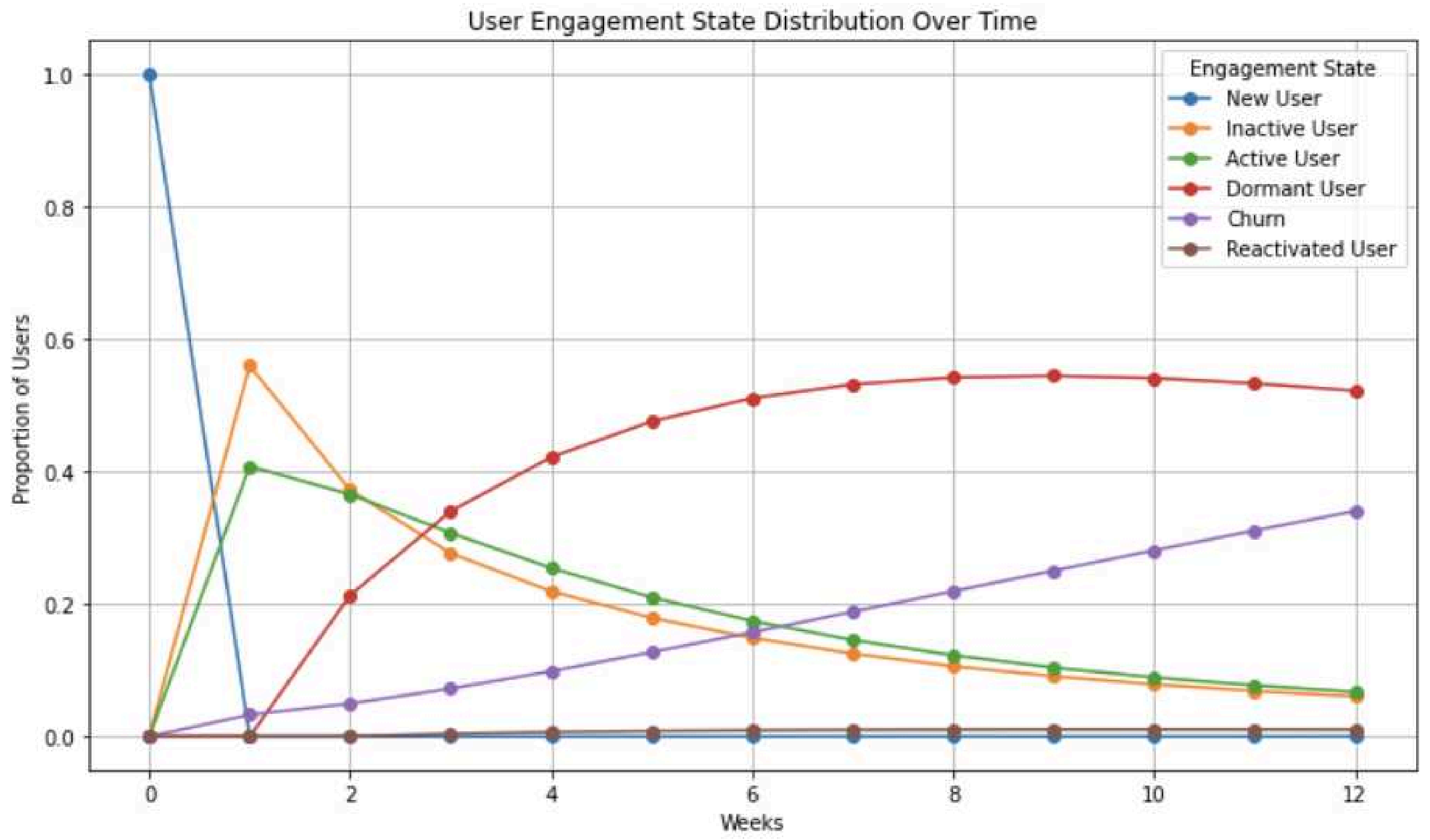
Built **multiple regression models** with:

- **Independent variables**:
  - User engagement metrics (e.g., started/completed workouts, app usage frequency).
- **Dependent variable**:
  - Subscription retention (e.g., active at Week 12).

Regression results showed that workout completions had the strongest relationship with users continuing their subscriptions. This highlights that users who complete workouts are significantly more likely to perceive value in the service and maintain their subscription over time.

Engagement States Research





Engagement States Research

Active state

We add 5% relative uplift
to needed transition

Active inactive	Active Active	Active Churn
0	0.05	0

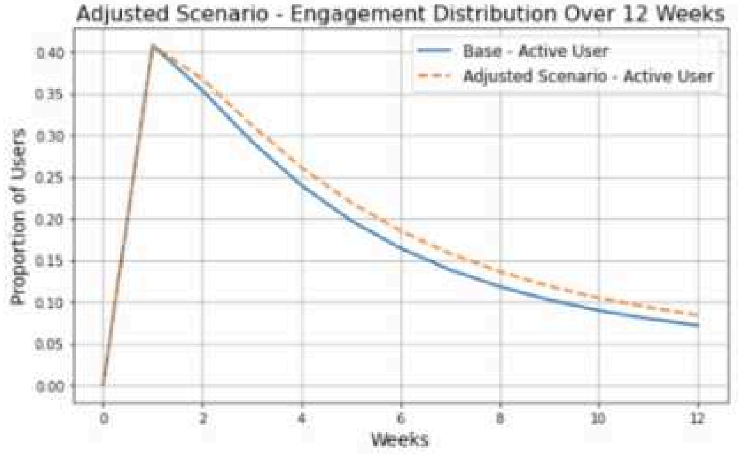
Dormant state

Dormant Active	Dormant Dormant	Dormant Churn
0	0	0

Reactivated state

Reactivated Inactive	Reactivated Active	Reactivated Churn
0	0	0

```
Adjusted transition matrix row for 'Active User' with uplift applied to 'Active User':
  - Uplift Applied: 0.05
```



Here is the uplift in
Active users by Nth week

	Base Active & Casual	Scenario Active & Casual	Change from Base (%)
Scenario			
Adjusted Scenario	0.0718	0.0842	17.26%

1 rows

Before uplift

After the uplift

Recommender System

Data Processing

1. Main **database** - PostgreSQL (AWS)
2. Main **DWH** - Snowflake, contains analytics and user data
3. Data in synced with **Airbyte** and processed with **Airflow** (Astronomer) + **dbt**
4. Training / inference data is put into **Feast** feature store

Fixed Library VS On-Demand Generation

Fixed Library

Set 1 REST 120sec
Chest 3sets

Superset 1 - 2 rounds REST 90sec
Chest
Back
Biceps

Superset 2 - 2 rounds REST 90sec
Chest
Shoulders
Triceps

Old approach

- Workout templates crafted by fitness expert
- Offline generation of workout DB + search algorithm
- Exercise DB - ~100 exercises
- Workout parameters - ~5 parameters with 3-10 dimensions - (intensity, fitness level, equipment, duration, body area)

Not scalable - adding new parameters results in exponential DB growth

On-Demand

Current approach

- Growing exercise library - 100 to 650 exercises
- Additional user parameters - 5 to 50+ (muscle freshness, health restrictions, sex, cardio machines, etc)
- Algorithm-based generation with ML components
- Future - using LLMs to generate workout templates

Fixed Library VS On-Demand Generation

Fixed library

+ Easier to control edge-cases

+ Can pre-compute ranking for some user parameters

- Long re-generation process

- Not scalable when number of parameters grows

On demand

+ Fast system updates

- Harder to integrate ranking models

Hybrid - generate multiple candidates on-demand and rank

Best of both worlds!

LLM workout plan generation

Workout plan is a schedule of workout parameters - workout type, target muscles, intensity, preferred rep range, etc.

Expert-crafted plans

Old approach

- Workout plans crafted by fitness expert for **4 predefined goals**
- **5** workout parameters defined
- 4 goals x 7 training frequencies = **28** plan templates

Not flexible enough - adding new parameters results in too much load on experts

LLM plans

Current approach

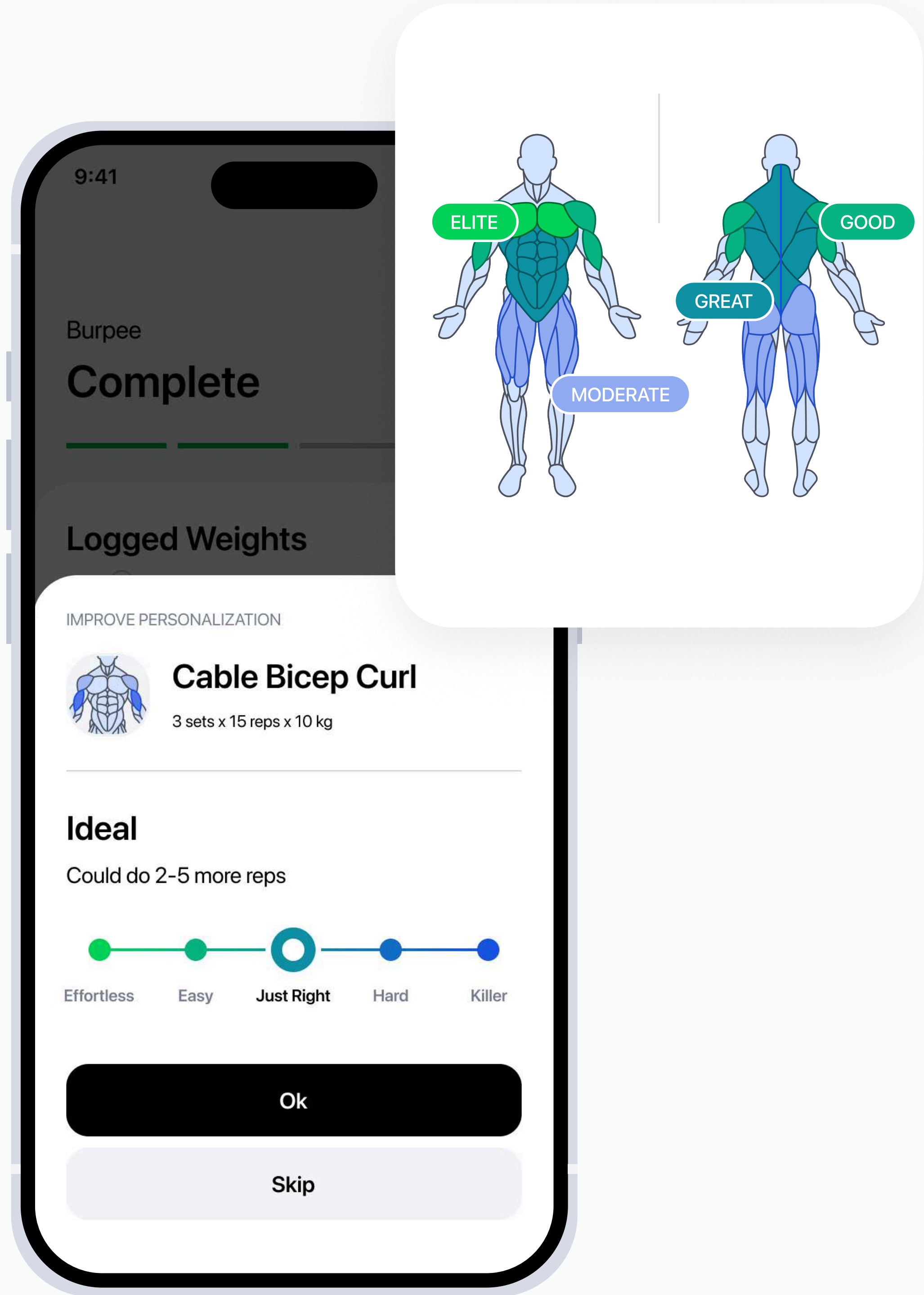
- LLM generates workout schedule with function call
- Any user goal is transformed into a set of parameters
- Easy to add new parameters, like preferred exercises

- Prompts are tested with regression test suite

- Next step - estimate performance and fine-tune

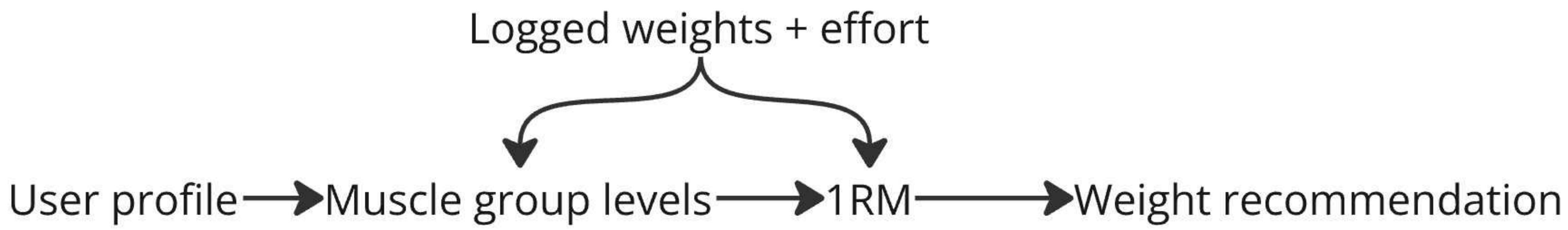
Weights Recommender

- This system recommends weights for exercises based on user's abilities and fitness level.
- We estimate level of each muscle group. We use onboarding data to predict the initial values, then continue to refine them from logged workouts.
- For each exercise, we estimate user's 1-rep max. This helps us track each user's progress and adopt weights for different protocols.



Weight recommendation - initial recommendation

Goal is to predict optimal weight for each exercise in a workout



Initial assessment - estimate "muscle group level" based on age, BMI, sex, activity level, fitness test results

Weight recommendation - initial recommendation

First version - manually tuned linear regression with hard-coded weights
After collecting user data - directly predict 1RM for each exercise

Labelled data - logged weights in first user-exercise session
Model - gradient boosting (catboost)

User features - BMI, age, sex, training frequency, fitness level, muscle group level
Exercise features - body area, target muscles, equipment

Clip model prediction to avoid over or under estimation

Result - +5% recommendation accuracy

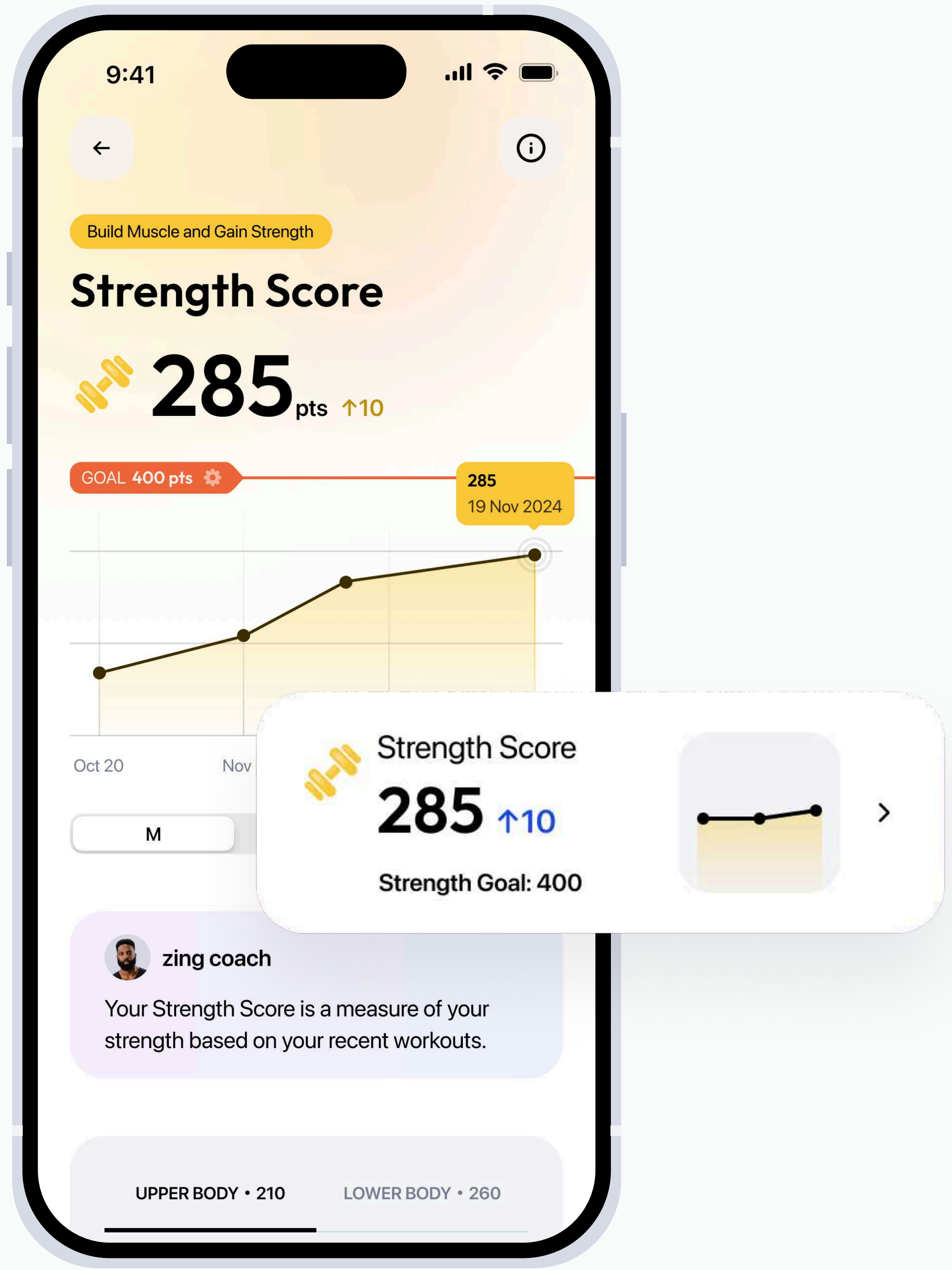
Weight recommendation — Strength Score

Strength Score is a metric that allows user to see how strong they are and how quickly they progress

Developed based on internal weight recommendation parameters - muscle group levels

It's useful to start from interpretable values, so they can be later exposed to the user

We also use it internally to estimate how good users reach their goals



System Monitoring and Evaluation

Offline evaluation

Regression testing

- Replay last 10000 workout generation attempts from production environment
- Measure workout and workout plans generation quality - muscle group coverage, equipment coverage, exercise variety, failed generation attempts

Simulators for sub-systems like weight recommendation

- Check basic invariants like “if user performs exercise with recommended parameters, weights are growing”

Regular reviews with fitness experts

System Monitoring and Evaluation

Online metrics

- Logs are uploaded to **Snowflake** for in-depth analytics
- **Hex** - automatic reports
- **Grafana** - basic metrics (error rates, performance and critical metrics like failed workout generation attempts)

Important for debugging - reproducible results - use seeded random number generators and log seed value

System Monitoring and Evaluation

Product metrics

- Evaluating uplift from workout recommendation improvements is hard as it’s usually small

- Our workout retention is up 25% YoY
- We estimate about 15% of improvements are attributed to workout recommendations

- It’s hard to run negative tests to measure true impact as we don’t have the scale of big tech and user feedback is really important for us

Takeaways

- **Start simple** - it’s possible to bring user value by using traditional algorithms and straightforward baselines
- **Think forward** - how will you measure system accuracy and collect user interaction to transition to ML-based solution
- **Invest into good evaluation pipeline** - it allows to significantly improve iteration speed and find a lot of bugs offline
- **LLMs help to improve** time to market significantly