وزارة الاتصالات وتكنولوجيا المعلومات

رواد مصر الرقمية

Recommendation System for FitnessandNutrition

Supervised by Eng. Mahmoud Talaat

Done by

Team members

Mariam Mahmoud

Hanya Wael

Eman Mostafa

Nour Mohamed

TABLE OF CONTENTS

01

Introduction

Introduction

Theglobal fitnessindustry is experiencing remarkable growth,drivenbyincreased awareness of health and wellness. However, many individuals continue to rely on generic, one-size-fits-all workout plans that fail to consider their unique physical conditions, goals, and preferences. This lack of personalization often leads to reduced motivation, inconsistent progress, and higher dropout rates. In this project, we have developed an **intelligent fitness recommendation system**that leverages **artificial intelligence and data analysis techniques**to generate **personalized workout** tailored to each user's profile. The aim of this system is to enhance user engagement, improve fitness outcomes, and promote healthier, more sustainable lifestyles by providing recommendations that truly fit individual needs.

02

Problem statement

Current Challenges in Achieving Personalized Fitness

Despite the rapid expansion of the fitnessindustry, most individuals still struggle to achieve their fitness goals due to several real-world challenges:

- 🏋️ Generic workout plansfail to consider users' individual goals, physical abilities, or health conditions —leading to ineffective results and loss of motivation.
- 📉 Lack of professional guidancemakes it difficult for beginners to
- design safe and efficient workout or nutrition routines.
- 📱 Limited personalization in existing fitness apps, which often recommend the same exercises regardless of user data or progress.
  💡 Difficulty tracking and adapting to user progress—many systems do not update plans dynamically as users improve or face challenges

03

Proposal solution

Intelligent Fitness Recommendation System

To address thelimitations oftraditional, one-size-fits-all workout approaches, we have developed an

🤖 AI-Based Fitness Recommendation System:

Our system provides personalized workout and nutrition guidance tailored to each user's unique profile. Users input key personal data — including age, weight, fitness goals, Experience level, and health conditions — and receive customized exercise routines designed specifically for them.

Using AI & data analytics, the system:

📊 Analyzes user data to suggest the most suitable workouts and progression schedules.

🔄 Dynamically updates recommendations as users improve or face new challenges, ensuring continuous progress.

The goal is to make fitness training smarter, safer, and more effective by providing a personalized, data-driven solution that enhances motivation, consistency, and long-term health outcomes.

To overcome the limitations of traditional one-size-fits-all approaches, we developed a two-phase system:

🏋️‍♂️ **Workout Type Prediction**: Predicts the user's most suitable workout type from six classes.

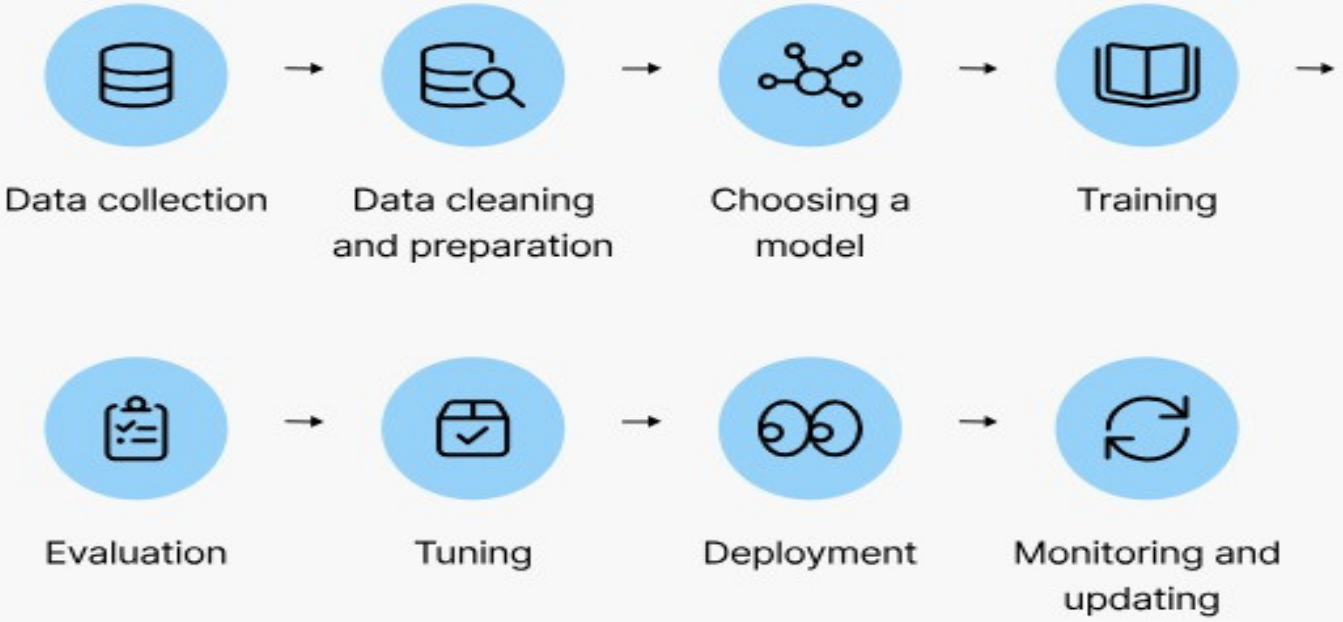
📌 **Content-Based Recommendation**: Generates a personalized workout routine based on the predicted type.

This approach ensures highly tailored, engaging, and results-oriented fitness guidance for every user.

04

Methodology

Methodology



Data collection

EXPLANATION FEATURES OF DATA: DATASET 1

- ❤️ Max_BPM: The maximum heart rate recorded during a workout session, indicating intensity level.
- 🩺 Avg_BPM: The average heart rate throughout the session, used to measure effort and endurance.
- 😌 Resting_BPM: The user's resting heart rate, reflecting cardiovascular fitness and recovery level.
- ⏱️ Session_Duration (hours): Indicates how long the workout session lasted, measured in hours.
- 🔥 Calories_Burned: The amount of energy (calories) burned during the workout session.
- 📅 Workout_Frequency (days/week): How many days per week the user typically exercises, used to gauge consistency.
- 🏋️ Experience_Level: Shows how advanced the user is (beginner(encoded to 1), intermediate(encoded to 2), advanced(encoded to 3)), guiding workout difficulty.
- 💧 Water_Intake (liters): The amount of water consumed daily in liters, used to assess hydration level.
- 🎯 Fitness_Goal: The user's main objective (weight loss, strength, endurance, muscle gain, etc.).
- 🧘 Workout_Type: The actual workout category performed (This is the ML prediction target.)
- 🏋️ Equipment Used (One-Hot Encoded):indicates whether user have this equipment or not

Data collection

EXPLANATION FEATURES OF DATA: DATASET 2

- 🦴 bodyPart: The primary body part targeted by the exercise (e.g., Chest, Back, Legs).
- 🏋️ equipment: The equipment used for the exercise (e.g., Dumbbell, Barbell, Bodyweight).
- 🎬 gifUrl: URL link to an animated GIF showing the exercise being performed.
- 🆔 id: Unique identifier for each exercise in the dataset.
- 🏷️ name: The name of the exercise (e.g., “Push-Up”, “Squat”).
- 🎯 target: The main muscle group worked by the exercise (e.g., Biceps, Quads, Chest).
- 💪 Secondary Muscles: Combines all secondary muscles (secondaryMuscles/0…/5) into a single column, showing additional muscles engaged by the exercise. Example: "Biceps, Forearms, Shoulders".
- 📝 Instructions (Full): Combines all instruction steps (instructions/0…/10) into a single column providing the complete step-by-step guide for performing the exercise.

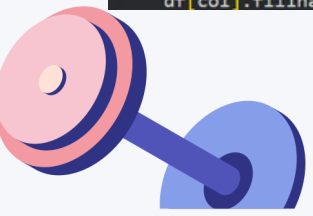
Data cleaning

for **users dataset** we replaced Nan values with median since it's robust to outliers and we filled missing values with 0 for all equipment-related columns

```
users.isna().sum()
```

```
Columns with missing values:
Fat_Percentage                120
Water_Intake (liters)          80
Equipment_dumbbell            190
Equipment_barbell             190
Equipment_cable               190
Equipment_kettlebell          190
Equipment_resistance_band     190
Equipment_stability_ball      190
Equipment_bodyweight_only     190
```

```
df['Fat_Percentage'].fillna(df['Fat_Percentage'].median(), inplace=True)
df['Water_Intake (liters)'].fillna(df['Water_Intake (liters)'].median(), inplace=True)

equipment_cols = ['Equipment_dumbbell', 'Equipment_barbell', 'Equipment_cable',
                  'Equipment_kettlebell', 'Equipment_resistance_band',
                  'Equipment_stability_ball', 'Equipment_bodyweight_only']

for col in equipment_cols:
    df[col].fillna(0, inplace=True)
```



Data analysis

In data analysis, we use key commands to understand the dataset before training. users.info() provides basic information about the columns, such as data types, non-null values, and overall structure. users.nunique() shows the number of unique values in each column, helping us identify whether a feature is useful for the model. Meanwhile, users.describe() gives a statistical summary of numerical columns, including the mean, median, standard deviation, and range, which helps us understand data distribution and detect outliers. All of this is essential for properly preparing the data before building a machine learning model.

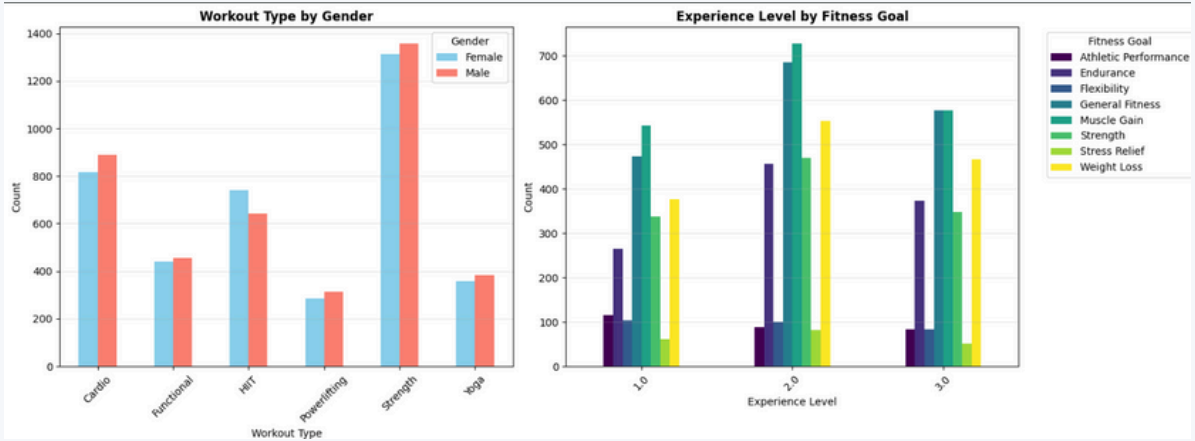
```
Data columns (total 23 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Age                           8000 non-null   float64
 1   Gender                        8000 non-null   object
 2   Weight (kg)                   8000 non-null   float64
 3   Height (m)                    8000 non-null   float64
 4   BMI                           8000 non-null   float64
 5   Fat_Percentage                7880 non-null   float64
 6   Max_BPM                       8000 non-null   float64
 7   Avg_BPM                       8000 non-null   float64
 8   Resting_BPM                   8000 non-null   float64
 9   Session_Duration (hours)      8000 non-null   float64
 10  Calories_Burned               8000 non-null   float64
 11  Workout_Frequency (days/week) 8000 non-null   float64
 12  Experience_Level              8000 non-null   float64
 13  Water_Intake (liters)         7920 non-null   float64
 14  Fitness_Goal                  8000 non-null   object
 15  Workout_Type                  8000 non-null   object
 16  Equipment_dumbbell            7810 non-null   float64
 17  Equipment_barbell             7810 non-null   float64
 18  Equipment_cable               7810 non-null   float64
 19  Equipment_kettlebell          7810 non-null   float64
 20  Equipment_resistance_band     7810 non-null   float64
 21  Equipment_stability_ball      7810 non-null   float64
 22  Equipment_bodyweight_only     7810 non-null   float64
dtypes: float64(20)  object(3)
```

```
users_df.nunique()
```

	count	mean	std	min	25%	50%	75%	max
Age	8000.0	34.723512	12.052514	16.00	25.7750	32.60	42.20	72.50
Weight (kg)	8000.0	68.868425	15.072511	40.00	58.6000	67.70	77.90	137.20
Height (m)	8000.0	1.706734	0.110925	1.45	1.6300	1.70	1.78	2.05
BMI	8000.0	23.739225	5.108056	10.00	20.6975	23.66	26.59	59.21
Fat_Percentage	7880.0	20.753680	5.780239	7.60	16.5000	20.60	24.70	45.00
Max_BPM	8000.0	184.169750	13.603001	140.00	175.0000	185.70	194.20	210.00
Avg_BPM	8000.0	141.818025	21.342400	90.00	126.5000	142.40	158.00	197.00
Resting_BPM	8000.0	63.043563	7.949840	45.00	57.1000	63.30	69.20	85.20
Session_Duration (hours)	8000.0	1.228191	0.436143	0.25	0.9000	1.19	1.51	2.50
Calories_Burned	8000.0	738.219300	371.992761	50.00	464.5750	694.15	979.00	2000.00
Workout_Frequency (days/week)	8000.0	3.737250	0.965231	1.00	3.0000	3.80	4.40	6.10
Experience_Level	8000.0	2.035250	0.776906	1.00	1.0000	2.00	3.00	3.00
Water_Intake (liters)	7920.0	2.943884	0.621191	1.50	2.5000	3.00	3.40	5.00
Equipment_dumbbell	7810.0	0.552113	0.497309	0.00	0.0000	1.00	1.00	1.00
Equipment_barbell	7810.0	0.459923	0.498423	0.00	0.0000	0.00	1.00	1.00
Equipment_cable	7810.0	0.424328	0.494272	0.00	0.0000	0.00	1.00	1.00
Equipment_kettlebell	7810.0	0.422151	0.493934	0.00	0.0000	0.00	1.00	1.00
Equipment_resistance_band	7810.0	0.389685	0.487755	0.00	0.0000	0.00	1.00	1.00
Equipment_stability_ball	7810.0	0.300256	0.458399	0.00	0.0000	0.00	1.00	1.00
Equipment_bodyweight_only	7810.0	0.748784	0.433740	0.00	0.0000	1.00	1.00	1.00

```
 0   bodyPart                1324 non-null   object
 1   equipment               1324 non-null   object
 2   gifUrl                  1324 non-null   object
 3   id                      1324 non-null   int64
 4   name                    1324 non-null   object
 5   target                  1324 non-null   object
 6   secondaryMuscles/0      1324 non-null   object
 7   secondaryMuscles/1      986 non-null    object
 8   instructions/0          1324 non-null   object
 9   instructions/1          1324 non-null   object
 10  instructions/2          1324 non-null   object
 11  instructions/3          1324 non-null   object
 12  instructions/4          1242 non-null   object
 13  instructions/5          739 non-null    object
 14  secondaryMuscles/2      233 non-null    object
 15  instructions/6          313 non-null    object
 16  instructions/7          92 non-null     object
 17  secondaryMuscles/3      32 non-null     object
 18  instructions/8          20 non-null     object
 19  secondaryMuscles/4      4 non-null      object
 20  instructions/9          5 non-null      object
 21  secondaryMuscles/5      2 non-null      object
 22  instructions/10         3 non-null      object
```

Users Dataset



Workout Type by Gender

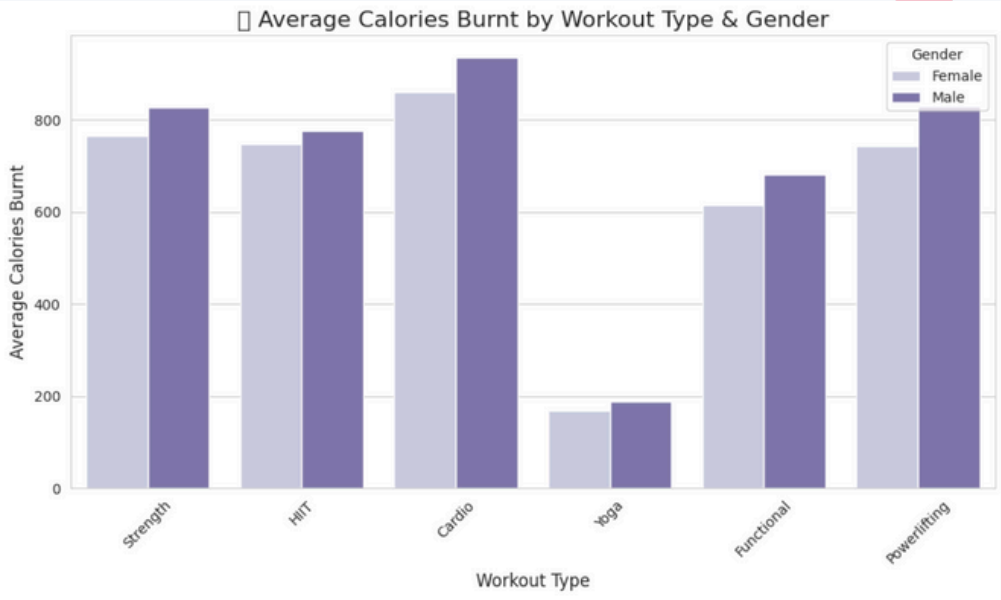
- Most Popular: Strength training is the clear favorite for both genders (~1,330 females, ~1,370 males).
- Gender Similarities: Participation is fairly balanced across most workout types.
- Least Popular: Powerlifting (~290 females, ~310 males) is the least chosen.
- Other Trends: Cardio is second (~820 females, ~900 males), HIIT third (~740 females, ~640 males).

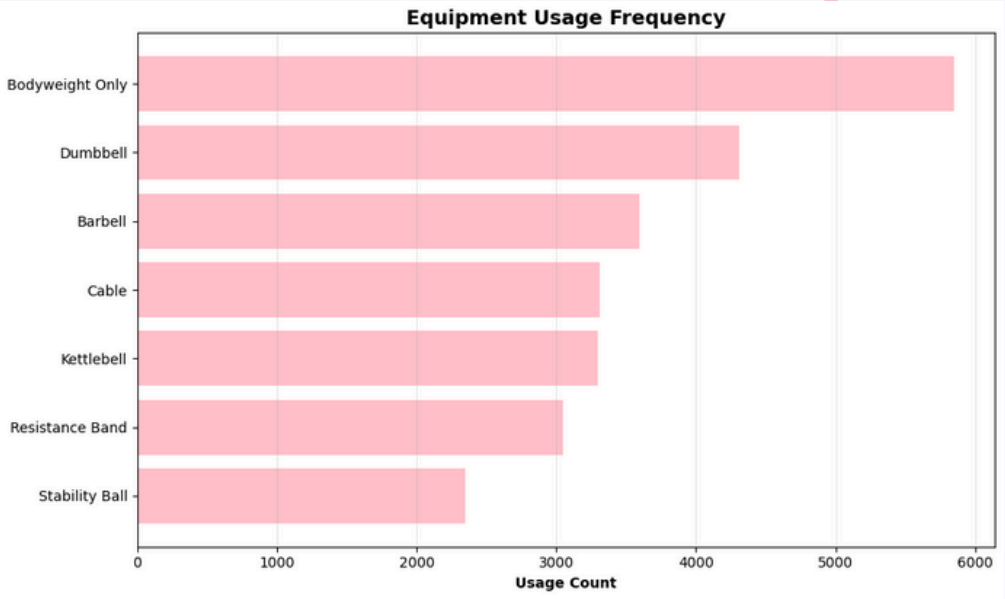
Experience Level by Fitness Goal

- General Fitness Leads: Most common goal across all levels, peaking at intermediate (2.0) with 700+ participants.
- Experience Distribution: Intermediate level (2.0) has the highest participation overall.
- Weight Loss: Popular among beginners (~380) and intermediates (~550), lower at advanced (~470).
- Specialized Goals: Athletic Performance and Flexibility remain niche across all levels.
- Advanced Focus: Muscle Gain rises as the second goal at advanced level (3.0, ~580), showing targeted strength focus.

Calories Burned by Workout Type and Gender:

- 👫 Gender: Males burn more calories than females; gap largest in Cardio, smallest in Yoga.
- 🔥 Top Workout: Cardio — highest calorie burn (M: ~930 kcal, F: ~870 kcal).
- 🧘 Lowest Workout: Yoga — lowest calories (M: ~190 kcal, F: ~165 kcal).
- 💪 Mid-Range: Strength, HIIT, Powerlifting → 750–830 kcal.
- ✅ Takeaway: Cardio = max calorie burn; Resistance training = good calorie burn + muscle-building.

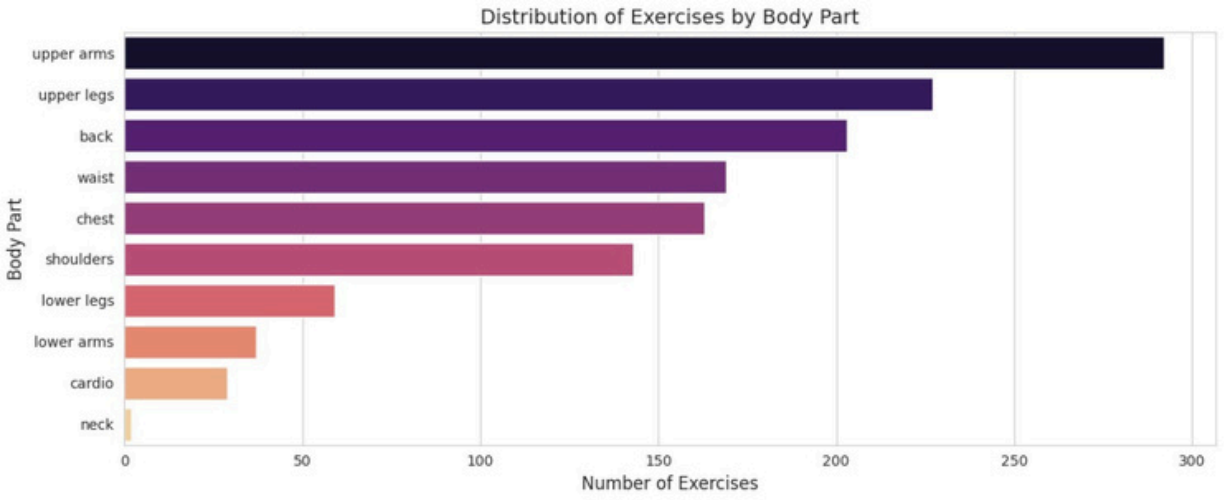




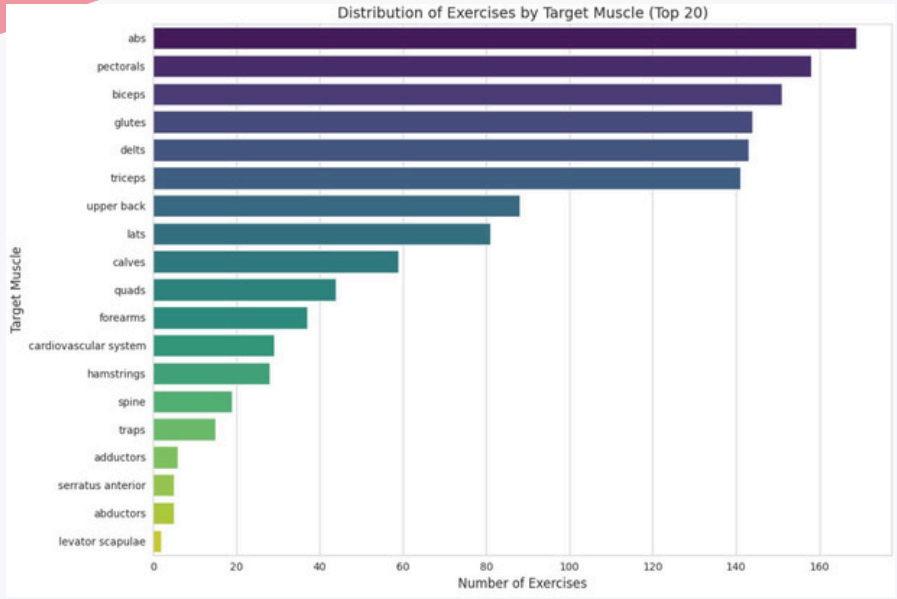
Equipment Popularity Rankings:

- Most Popular: Bodyweight exercises dominate (~5,850 uses), showing preference for accessible, no-equipment workouts.
- Top Free Weights: Dumbbells (~4,300) and barbells (~3,600) follow, highlighting traditional strength training's importance.
- Mid-Tier: Cable machines, kettlebells, and resistance bands (~3,000–3,300 uses) show moderate, similar popularity.
- Least Used: Stability balls (~2,350 uses) are far less common.

Exercises Dataset



- Dataset Focus: The dataset is overwhelmingly focused on strength training, particularly for the upper body.
- Highest Exercise Variety: "Upper arms" has the most exercise variety by a wide margin (nearly 300 exercises), followed by "upper legs" (approx. 235) and "back" (approx. 210).
- Moderate Variety: The "waist" (core), "chest", and "shoulders" have a solid, moderate number of exercises available (ranging from ~145 to 165).
- Significant Gaps: There is a sharp drop-off in exercise availability for "lower legs" (~60), "lower arms" (~40), and cardio (~30).
- Negligible Category: "Neck" exercises are almost non-existent in this dataset.

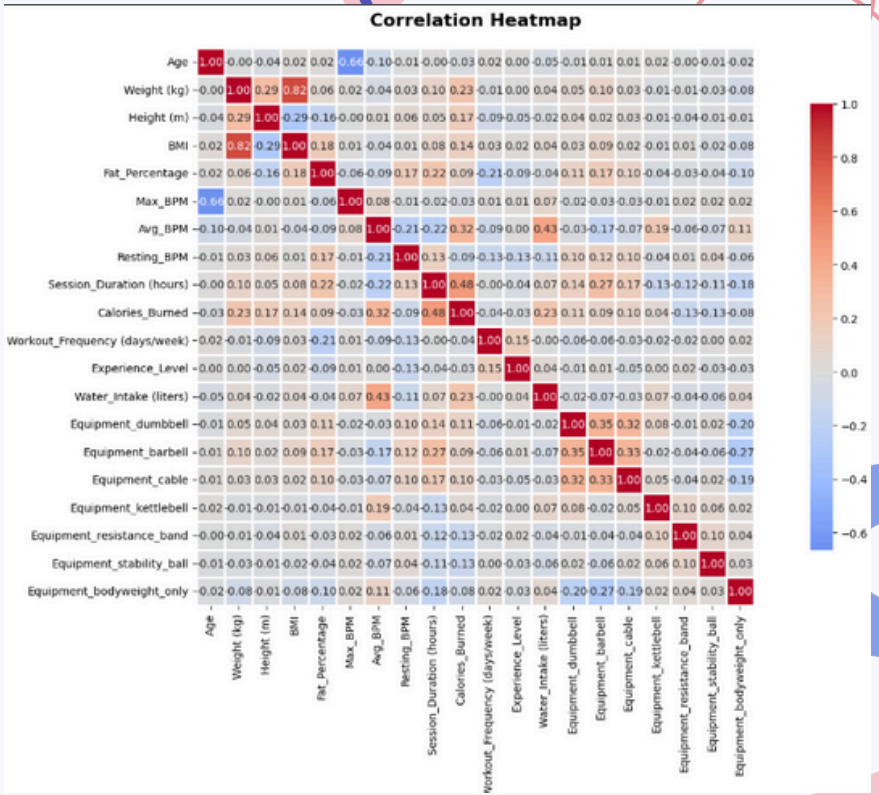


Exercise Variety & Muscle Focus

- Highest Variety: Core and "push" muscles dominate, with abs (~165) and pectorals (~155) leading, followed by shoulders and triceps (~140–150 exercises).
- Leg Imbalance: Quads and glutes are well-represented, but hamstrings (~25) and some other leg muscles have far fewer exercises.
- Back Muscle Split: Upper back (~95) and lats (~85) have decent coverage, while traps and spine-focused exercises are limited.
- Low Focus Areas: Cardio (~30) and forearms (~35) are underrepresented, emphasizing resistance training over general conditioning.
- Niche Muscles: Adductors, serratus anterior, abductors, and levator scapulae have very few exercises, forming the bottom of the dataset.

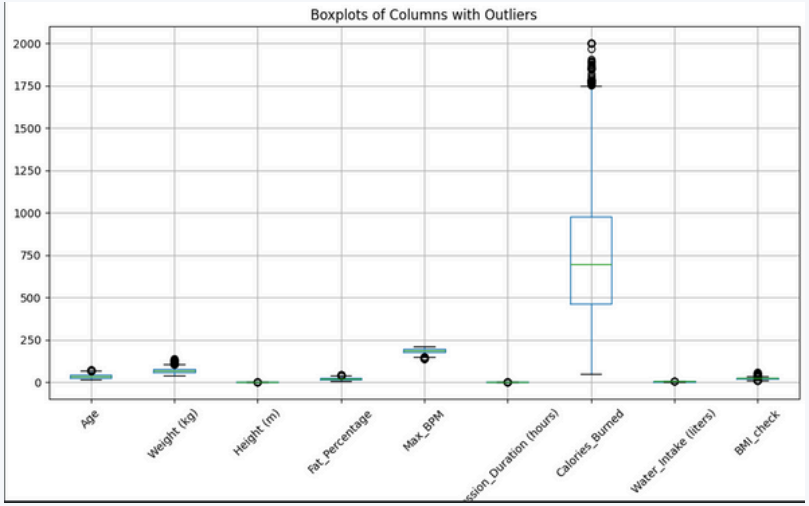
Correlation Analysis:

- 💪 Physical & Heart Metrics: Weight ↔ BMI (0.92), Max_BPM ↔ Avg_BPM (0.86)
- 🏋️ Equipment Patterns: Dumbbell ↔ Barbell (0.35), Resistance Band ↔ Stability Ball (0.32)
- 📉 Negative Trends: Longer sessions → lower body fat (-0.25); slight cardio-strength trade-offs
- 👶 Weak/No Correlations: Age, Workout Frequency, Experience Level mostly independent
- 🔥 Takeaway: Fitness outcomes are multifactorial → focus on multiple factors rather than one metric

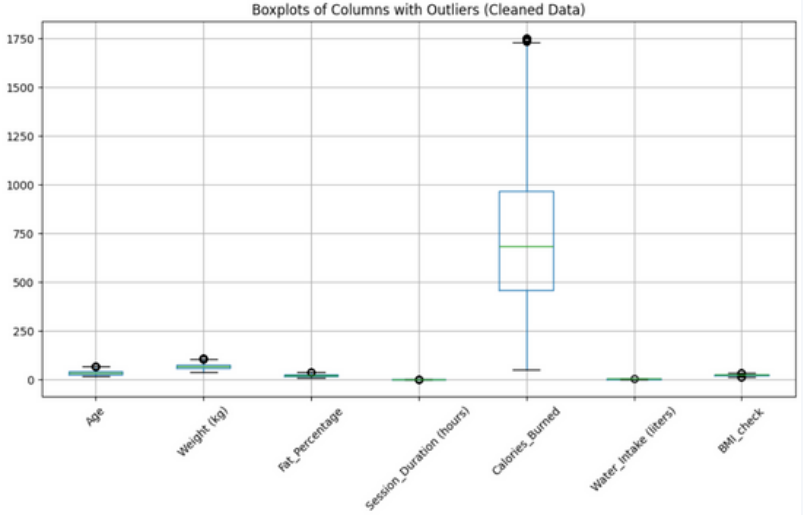


Outlier Detection & Handling

before



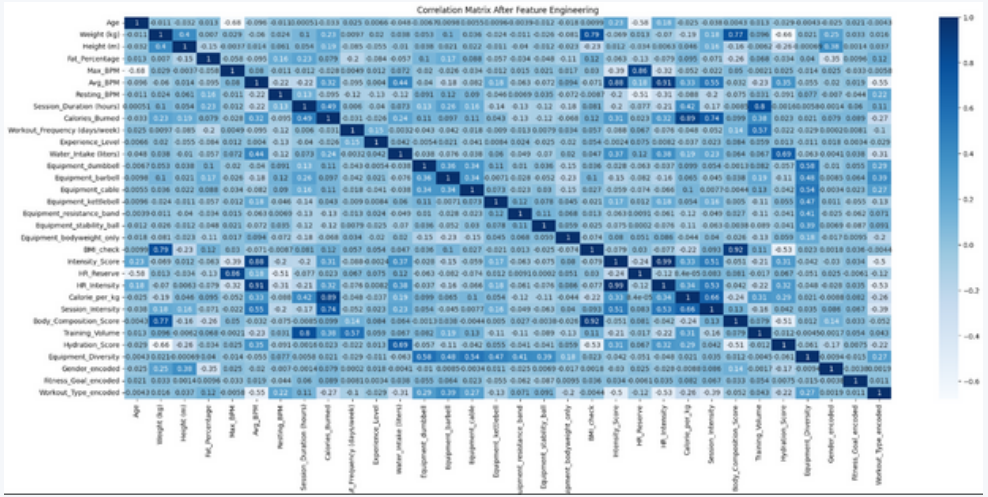
after



we kept just the outliers that make sense

User Data Preprocessing & Feature engineering

```
df['Intensity_Score'] = (df['Avg_BPM'] / df['Max_BPM']) * 100
# Heart Rate Features
df['HR_Reserve'] = df['Max_BPM'] - df['Resting_BPM']
df['HR_Intensity'] = (df['Avg_BPM'] - df['Resting_BPM']) / (df['HR_Reserve'] + 0.1)
df['HR_Intensity'] = df['HR_Intensity'].clip(0, 1)
# Fitness Indicators
df['Calorie_per_kg'] = df['Calories_Burned'] / (df['Weight (kg)'] + 1)
df['Session_Intensity'] = df['Calories_Burned'] / (df['Session_Duration (hours)'] + 0.1)
df['Body_Composition_Score'] = df['BMI_check'] * (1 - df['Fat_Percentage']/100)
# Training Characteristics
df['Training_Volume'] = df['Workout_Frequency (days/week)'] * df['Session_Duration (hours)'
df['Hydration_Score'] = df['Water_Intake (liters)'] / (df['Weight (kg)'] * 0.03)
df['Equipment_Diversity'] = df[equipment_cols].sum(axis=1)
```



We created 10 new features to enhance the fitness dataset, focusing on performance, efficiency, and body composition. Key metrics include Intensity Score and HR Reserve/HR Intensity for cardiovascular performance, Calorie per kg and Session Intensity for workout efficiency, and a Body Composition Score combining BMI and body fat percentage to reflect overall fitness

Exercise Data Preprocessing & Feature Engineering

Data Cleaning & Consolidation:

- Combined multiple secondary muscle columns into a single secondary_muscles column.
- Merged multiple instruction columns into instructions_full for complete steps and instructions_short for the first 3 steps.

Enhanced Muscle Group Mapping

- Standardized target muscles into broader groups (e.g., Chest, Back, Shoulders, Core).
- Categorized muscles by size: Big (Chest, Back, Quads, Hamstrings) vs Small (Shoulders, Biceps, Triceps, etc.).

Equipment Difficulty & Encoding

- Assigned a difficulty score based on equipment type (Body weight = Easy, Dumbbells = Moderate, Barbells = Hard).
- Encoded categorical features (equipment and muscle_group) for modeling using label encoding.

```
    difficulty_mapping = {
        'body weight': 1, 'assisted': 1, 'band': 1, 'resistance band': 1,
        'dumbbell': 2, 'kettlebell': 2, 'medicine ball': 2, 'stability ball': 2,
        'cable': 2, 'leverage machine': 2, 'rope': 2, 'bosu ball': 2,
        'barbell': 3, 'ez barbell': 3, 'olympic barbell': 3, 'smith machine': 3,
        'trap bar': 3, 'sled machine': 3, 'weighted': 3
    }
    clean_df['difficulty_level'] = clean_df['equipment'].map(difficulty_mapping).fillna(2)

    # Exercise type mapping
    exercise_type_mapping = {
        'waist': 'Core', 'back': 'Strength', 'chest': 'Strength',
        'upper arms': 'Strength', 'shoulders': 'Strength', 'upper legs': 'Strength',
        'lower legs': 'Strength', 'lower arms': 'Strength', 'cardio': 'Cardio',
        'neck': 'Flexibility'
    }
    clean_df['exercise_type'] = clean_df['bodyPart'].map(exercise_type_mapping)

    # Accessibility score (1=gym only, 5=home friendly)
    accessibility_mapping = {
        'body weight': 5, 'band': 5, 'resistance band': 5,
        'dumbbell': 4, 'kettlebell': 4, 'medicine ball': 4,
        'stability ball': 3, 'cable': 2, 'barbell': 2,
        'smith machine': 1, 'sled machine': 1, 'leverage machine': 1
    }
    clean_df['accessibility'] = clean_df['equipment'].map(accessibility_mapping).fillna(3)

    # Calculate complexity score based on instruction length
    clean_df['complexity'] = clean_df['instructions'].str.len() / 100
    clean_df['complexity'] = clean_df['complexity'].clip(1, 5)

    return clean_df

# Clean exercise dataset
exercises = clean_exercises_dataset(exercises_raw)
print(f"✅ Exercise dataset cleaned: {exercises.shape}")

# Display basic statistics
print("\n📊 DATASET OVERVIEW")
print("-" * 30)
print("User Dataset Features:")
```

Exercise Data Preprocessing & Feature Engineering

Feature Vector Creation

- Each exercise was converted into a structured feature vector including: bodyPart, target, equipment, difficulty_score, and muscle_group.
- Ensures each exercise can be compared numerically for similarity computation.

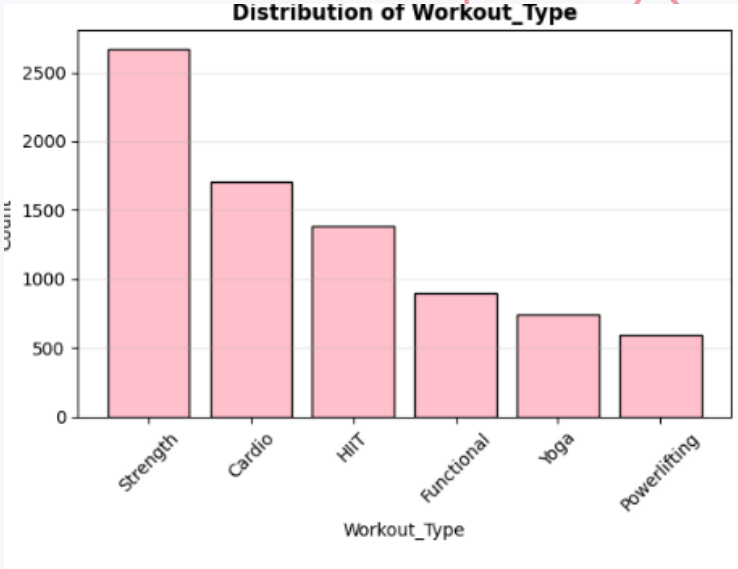
Categorical Feature Encoding

- One-hot encoded categorical features (body Part, target, equipment, muscle_group).
- Combined with numeric difficulty_score to create a unified feature matrix.
- Standardized numeric features (0–1 range) for consistent similarity calculations.

```
# Combines secondary muscles, creates full + short instructions

# 1. Copy original DataFrame
exercises_clean = exercises_df.copy()

# 2. Combine secondary muscles into one column
secondary_cols = [col for col in exercises_clean.columns if 'secondaryMuscles' in col]
exercises_clean['secondary_muscles'] = exercises_clean[secondary_cols].apply(
    lambda row: ', '.join([str(val) for val in row if pd.notna(val) and val != 'NaN']),
    axis=1
)

# 3. Combine ALL instruction steps into 'instructions_full'
instruction_cols = [col for col in exercises_clean.columns if 'instructions' in col]
exercises_clean['instructions_full'] = exercises_clean[instruction_cols].apply(
    lambda row: ' '.join([str(val) for val in row if pd.notna(val) and val != 'NaN']),
    axis=1
)

# 4. Create 'instructions_short' with first 3 steps
def get_first_3_instructions(row):
    steps = []
    for i in range(3):
        col = f'instructions/{i}'
        if col in row.index and pd.notna(row[col]) and row[col] != 'NaN':
            steps.append(f"{i+1}. {row[col]}")
    return "\n".join(steps) if steps else "No instructions available"

exercises_clean['instructions_short'] = exercises_clean.apply(get_first_3_instructions, axis=1)

# 5. Keep only relevant columns (plus gifUrl)
exercises_clean = exercises_clean[[
    'id', 'name', 'bodyPart', 'target', 'equipment',
    'secondary_muscles', 'instructions_full', 'instructions_short'
]]

# 6. Verify results
print("Preprocessed exercises:", exercises_clean.shape)
print("Sample full instructions:\n", exercises_clean['instructions_full'].iloc[0])
print("Sample short instructions:\n", exercises_clean['instructions_short'].iloc[0])
```

Workout Type Distribution(Target):

- **Strength** is the most recorded workout, dominating the dataset.
- **Cardio and HIIT** follow, showing strong user interest in endurance and high-intensity training.
- **Functional workouts** appear at a moderate level.
- **Yoga and Powerlifting** are the least frequent categories.
- **Overall,** the data shows clear class imbalance, which must be handled during modeling to avoid biased predictions.



imbalanced Data Handling

During data exploration, we observed that the distribution of fitness categories—such as workout types or nutrition levels—was highly imbalanced. Some categories contained significantly more samples than others. This imbalance can cause the model to favor majority classes, leading to biased predictions and lower accuracy for underrepresented groups.

```
Class distribution BEFORE SMOTE:
Cardio: 1637 (21.2%)
Functional: 892 (11.5%)
HIIT: 1334 (17.2%)
Powerlifting: 599 (7.7%)
Strength: 2542 (32.8%)
Yoga: 735 (9.5%)
```

Techniques Used to Address Imbalance

✅SMOTE (Synthetic Minority Oversampling Technique):

- Generates new, synthetic samples for minority classes rather than simply duplicating existing ones.
- Creates these samples by interpolating between nearest neighbors, helping the model learn smoother decision boundaries.
- Ensures that all fitness categories are represented more evenly in the training data, improving fairness and model performance

```
Class distribution AFTER SMOTE:
Cardio: 2033 (26.3%)
Functional: 2033 (26.3%)
HIIT: 2033 (26.3%)
Powerlifting: 2033 (26.3%)
Strength: 2033 (26.3%)
Yoga: 2033 (26.3%)
```

05

Machine Learning Results (phase one)

Models Training

To find the best model for predicting workout types, we tried several tree-based classifiers:

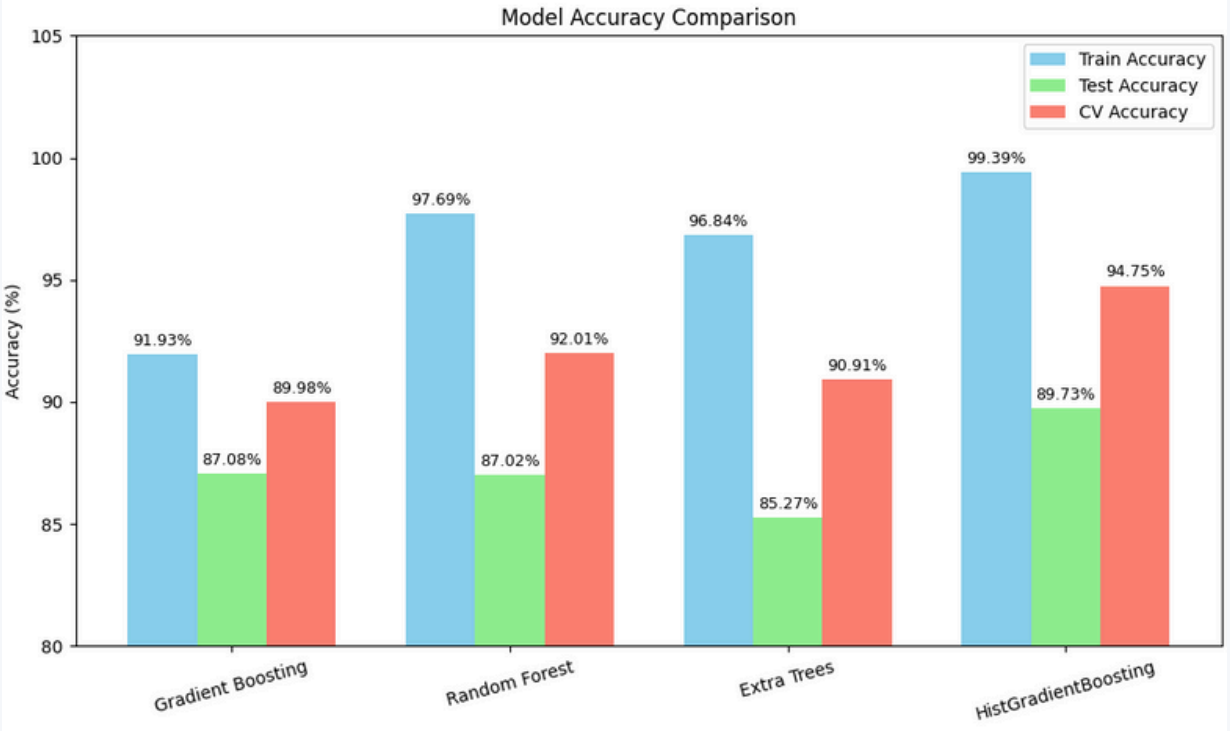
- ⚡ Gradient Boosting
- 🌲 Random Forest
- 🌳 Extra Trees
- 🚀 HistGradientBoosting

Each model was trained on the same data, and we evaluated them using:

- ✅ Train Accuracy
- 📊 Test Accuracy
- 🔄 5-Fold Cross-Validation Accuracy

MODELS COMPARISON

- HistGradientBoosting demonstrates the best generalization (highest CV score)
- All models show some overfitting (train > test), but within acceptable ranges
- HistGradientBoosting achieves the best bias-variance tradeoff
- Test accuracies range from 85-90%, indicating robust real-world performance



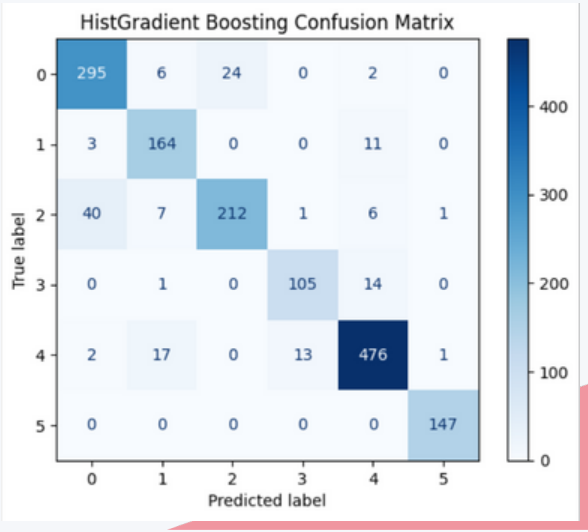
Model Selection

Why We Selected HistGradientBoosting?

Based on our comprehensive model evaluation, HistGradientBoosting was selected as the optimal model for predicting fitness experience levels due to its superior performance across all key metrics:

CONFUSION MATRIX:

- the model correctly classifies most samples across all 6 classes, showing strong performance overall.✅
- **Classes 4 and 5** have the highest accuracy with very few errors.
- **Classes 0 and 1** also perform well with minimal misclassifications.
- **Class 2 and Class 3** show moderate confusion with neighboring classes, especially 0, 1, and 4.
- **Misclassifications** are relatively small compared to correct predictions, meaning the model generalizes well despite class imbalance.



Feature Importance

Top Predictors of Workout type:

Fitness_Goal_encoded (~19%)

- -Most influential feature
- -User goals (weight loss, muscle gain, endurance) strongly indicate experience
- -Shuffling causes largest performance drop

Avg_BPM (~8%)

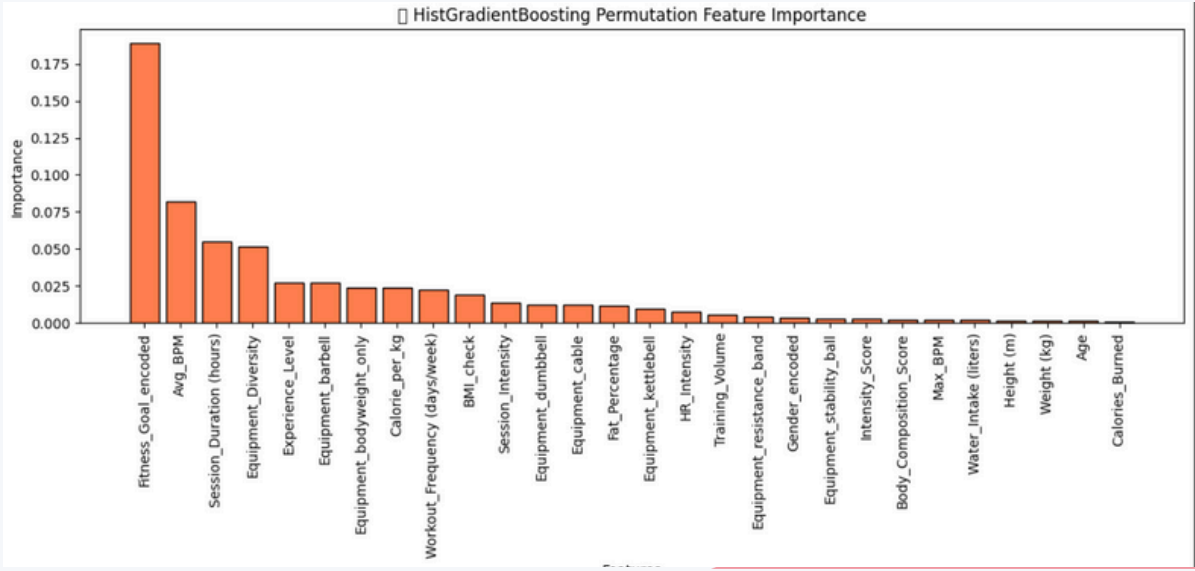
- Average heart rate during workouts
- Reflects cardiovascular fitness and workout intensity

Session_Duration (hours) (~6%)

- Longer training sessions usually indicate higher experience
- Captures commitment and patterns of users

Equipment_Diversity (~5%)

- Variety of equipment used shows adaptability and



06

Unsupervised Model (phase 2)

Content-Based Filtering pipeline

Content-Based Filtering (Exercise Recommendation)

Input: Target Muscle Group + User Constraints (injuries, equipment)

↓

Feature Vectorization (1,324 exercises × 72 features each)

↓

Cosine Similarity Matrix (1,324 × 1,324)

↓

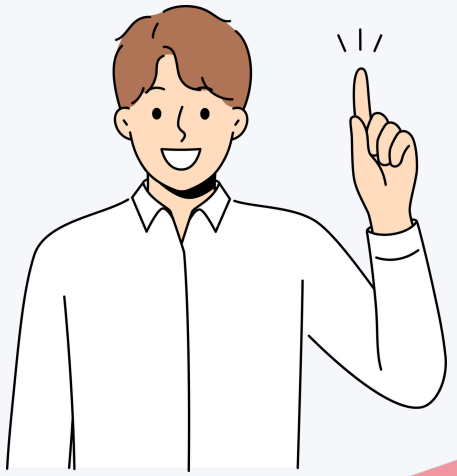
Injury Filtering (removes unsafe exercises)

↓

Output: Top N exercises per muscle group

↓

Diversity Filter (ensures variety - not 3 curl variations)



Content-Based Filtering pipeline

Personalization Engine

Input: Predicted workout type + User goals/injuries

↓

Generate 5-day split plan with sets/reps based on goal

↓

Calculate macros (Mifflin-St Jeor formula)

↓

Sleep recommendations (based on training volume)

↓

Output: Complete workout plan with nutrition

Content-Based Filtering pipeline

Content-Based Filtering (Exercise Recommendation)

Input: Target Muscle Group + User Constraints (injuries, equipment)

↓

Feature Vectorization (1,324 exercises × 72 features each)

↓

Cosine Similarity Matrix (1,324 × 1,324)

↓

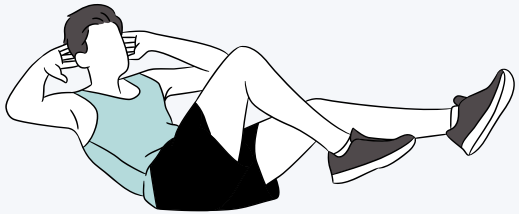
Injury Filtering (removes unsafe exercises)

↓

Output: Top N exercises per muscle group

↓

Diversity Filter (ensures variety - not 3 curl variations)



CONTENT-BASED FILTERING

How does it work?

Each exercise is converted to a 72-dimensional feature vector

- Each categorical feature gets one-hot encoded (turns into 0s and 1s)
- One-hot encoding = dumbbell bench press becomes (1,0,0,...,0) for dumbbell, (1,0,0,...,0) for barbell category, etc.
- Captures all exercise attributes in a format computers can compare

We compute a similarity matrix using cosine similarity

SIMILARITY DISTRIBUTION BY RANGE

Range	Count	Percentage	Visual
0.0-0.1	211,652	12.07 %	
0.1-0.2	43,428	2.48 %	
0.2-0.3	37,262	2.13 %	
0.3-0.4	14,448	0.82 %	
0.4-0.5	18,888	1.08 %	
0.5-0.6	26,704	1.52 %	
0.6-0.7	29,000	1.65 %	
0.7-0.8	39,536	2.26 %	
0.8-0.9	2,868	0.16 %	
0.9-1.0	11,560	0.66 %	

CONTENT-BASED FILTERING

Cosine Similarity:

What does this number mean?

- 1.0 = identical exercises (same muscle, equipment, difficulty)
- 0.65 = very similar (dumbbell bench press vs machine chest press)
- 0.0 = completely different muscle groups
- -0.23 = opposite characteristics (rare)

Key Insights:

- 87.53% of similarities are NEGATIVE or near-zero → exercises are diverse
- Only 4.07% have similarity >0.6 → true duplicates are rare
- We can use 0.65 threshold to recommend similar yet diverse exercises

Unsupervised Pipeline

automatic target muscle group assignment

- INPUT: User specifies training frequency (3/4/5 days/week)

↓

- SYSTEM AUTO-GENERATES MUSCLE SPLIT

```
workoutsplitmap = {
    3: {
        1: ["Chest", "Triceps"],
        2: ["Back", "Biceps"],
        3: ["Quads", "Hamstrings", "Calves", "Glutes", "Core"],
    },
    4: {
        1: ["Chest", "Triceps"],
        2: ["Back", "Biceps"],
        3: ["Quads", "Hamstrings", "Calves"],
        4: ["Shoulders", "Forearms"],
    },
    5: {
        1: ["Chest", "Triceps"],
        2: ["Back", "Biceps"],
        3: ["Quads", "Hamstrings", "Calves"],
        4: ["Shoulders", "Forearms"],
        5: ["Glutes", "Core"],
    },
}
```

CONTENT-BASED FILTERING

Equipment Filtering :

Filter by equipment available:
├── If Gym: Keep all (barbell, dumbbell, machine, cable, etc.)
├── If Home: equipment_map = {
'a': ['body weight', 'band'], #Minimal (body weight, resistance band only)
'b': ['body weight', 'band', 'dumbbell'], #Basic (+ dumbbell)
'c': ['body weight', 'band', 'dumbbell', 'kettlebell', 'medicine ball', 'stability ball']
, #Standard (+ kettlebell, medicine ball, stability ball)
'd': ['body weight', 'band', 'dumbbell', 'kettlebell', 'medicine ball', 'stability ball','cable'
], #Full home setup
}
├── If Both: Keep all ↓ Example (Home setup b):
└── 144 → 67 exercises remain

Difficulty Filtering (Experience Match):

67 chest exercises (after equipment filter)
Match difficulty to experience:
├── Beginner (Level 1): Only difficulty 1 exercises
├── Intermediate (Level 2): Difficulty 1-2 exercises
├── Advanced (Level 3): All difficulties 1-3
Example (Intermediate): ---->Keep exercises with difficulty ≤ 2

Injury Handler (CRITICAL SAFETY STEP)

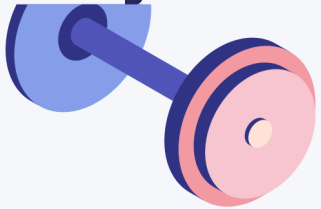
Injury Filtering :

After filtering by muscle, equipment, and difficulty, we still have exercises that could harm users with injuries.

```
INPUT:
├─ Exercise pool (e.g., 94 safe chest exercises)
├─ User injuries (e.g., ["shoulder injury", "lower back pain"])
└─ INJURY_KEYWORDS dictionary

PROCESS:
    for injury in user_injuries:
        keywords = INJURY_KEYWORDS[injury]

        for exercise in exercise_pool:
            # Check BOTH exercise name AND target muscle
            if any(keyword in exercise.name) OR
               any(keyword in exercise.target):

                ✗ REMOVE exercise from pool

                Example:
                └─ "Overhead Press" contains "overhead"
                   (matches shoulder keyword)
                   → REMOVED

    return filtered_safe_exercises
```

```
Starting pool: 144 chest exercises
    ↓
Keyword matching for "shoulder", "overhead", "press", etc.
    ├─ "Dumbbell Bench Press" → Contains "press" BUT NOT overhead
    │  └─ ⚠ RISKY - might stress shoulder
    ├─ "Overhead Dumbbell Press" → Contains "overhead"
    │  └─ ✗ REMOVED (too dangerous)
    ├─ "Cable Fly" → No shoulder keywords
    │  └─ ✓ SAFE
    ├─ "Machine Chest Press" → Contains "press"
    │  └─ ⚠ RISKY - rechecked manually
    └─ "Dumbbell Floor Press" → No shoulder keywords
       └─ ✓ SAFE
    ↓
Result: 94 safe chest exercises remain (from 144)
Removed: 50 exercises (34.7%)
```

SIMILARITY-BASED SELECTION (NOW WITH FILTERS APPLIED)



```text
1. Pick first exercise from filtered pool
   └─ "Dumbbell Bench Press" (seed exercise)

2. Look up its similarity scores from matrix
   Row 127: [1.0, 0.92, 0.78, 0.65, 0.32, ...]

3. Find matches in FILTERED POOL (only the 67 safe ones)
   ├─ Match 1: "Machine Chest Press" (0.92) ✓ In safe pool
   ├─ Match 2: "Dumbbell Fly" (0.78) ✓ In safe pool
   ├─ Match 3: "Cable Chest Press" (0.65) ✓ In safe pool
   ├─ Match 4: "Smith Chest Press" (0.60) ✗ Below threshold
   └─ (Stop searching once found enough)

4. Apply similarity threshold (0.65)
   ├─ Keep if similarity ≥ 0.65
   ├─ Discard if similarity < 0.65
   └─ Result: Top 3-4 most similar, safe exercises
```

DIVERSITY FILTERING 'Final step' :

Problem: After similarity selection, all 4 exercises might be:

- All dumbbells (same equipment = monotonous)
- All press variations (same movement family = boring)

We handled it by family+equipment filter

WARMUP & COOLDOWN PROTOCOLS

Warmups and cooldowns are critical components of training that:

- Warmup: Prepares body, raises heart rate, activates target muscles, prevents injury
- Cooldown: Aids recovery, reduces heart rate gradually, improves flexibility, prevents dizziness

our system automatically generates personalized warmups and cooldowns based on the day's muscle groups. ‘predifined logic’

SYSTEM OUTPUT (demo)

```
...  ================================================================================
                          YOUR PERSONALIZED WORKOUT PLAN
     ================================================================================

                                  PROGRAM DETAILS
     --------------------------------------------------------------------------------
     Age:                    26
     Gender:                 Female
     BMI:                    23.5
     Training Days:          5/week
     Experience Level:       Intermediate
     Fitness Goal:           WEIGHT LOSS
     Training Location:      Gym
     Injuries Managed:       shoulder injury, lower back pain


     ================================================================================
                          DAY 1 - Chest, Triceps (7 EXERCISES)
     ================================================================================

                                  WARM-UP (5-7 minutes)
     --------------------------------------------------------------------------------
     1. Light Cardio - 3-5 minutes
        └ Jumping jacks, jogging in place, or cycling
     2. Arm Circles - 30s each direction
        └ Large circles forward/backward
     3. Band Pull-Aparts - 15 reps
        └ Light band, squeeze shoulder blades

                                  MAIN WORKOUT
     --------------------------------------------------------------------------------


     1. CABLE ONE ARM FLY ON EXERCISE BALL
        Muscle Group: Chest | Target: pectorals
        Equipment: cable
        Sets: 3 | Reps: 12-20 | Rest: 45s
        Instructions: Follow standard form

     2. CABLE ONE ARM LATERAL BENT-OVER
        Muscle Group: Chest | Target: pectorals
        Equipment: cable
        Sets: 3 | Reps: 12-20 | Rest: 45s
        Instructions: Follow standard form
```

PERSONALIZED NUTRITION (MACROS)

Algorithm: Mifflin-St Jeor Formula

- Generic calorie estimates are inaccurate. We need precision nutrition tied to the user's specific profile.
- Solution: Industry-standard Mifflin-St Jeor formula

Calculate BMR (Basal Metabolic Rate)

Formula varies by gender:

For Females:

$$BMR = 10 \times weight + 6.25 \times height - 5 \times age - 161$$

For Males:

$$BMR = 10 \times weightkg + 6.25 \times heightcm - 5 \times age + 5$$

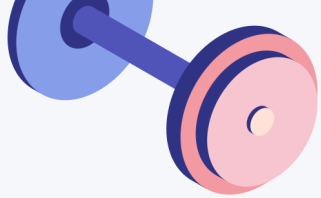
Calculate TDEE (Total Daily Energy Expenditure)

Formula:

$$TDEE = BMR \times Activity\ Multiplier$$

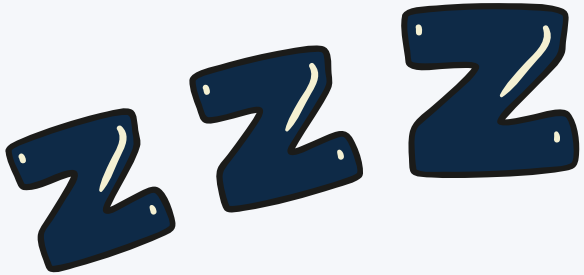
```
NUTRITION RECOMMENDATIONS

Daily Calorie Targets:
├─ BMR: 1,594 kcal (metabolism at rest)
├─ TDEE: 2,471 kcal (with activity)
└─ Target: 1,971 kcal (500 deficit for weight loss)

Daily Macro Targets:
├─ Protein: 167g (30% of calories)
├─ Carbs: 207g (44% of calories)
└─ Fat: 46g (26% of calories)
```

SLEEP RECOMMENDATIONS



Training Volume-Based Sleep Formula

- Generic "8 hours" doesn't account for training load. Heavy training needs more recovery.
- Solution: Dynamic sleep recommendation tied to training volume, experience, and goal



Base Sleep Calculation

Formula:

$$\text{Sleep Hours} = 7.5 + (\text{Volume Factor}) + (\text{Experience Factor}) + (\text{Goal Factor})$$

Training Volume Adjustment

Formula:

$$\text{Volume Factor} = \text{Training Days} \times 0.2 \text{ hours}$$

Logic: More training days = more muscle damage = need more recovery

Experience Level Adjustment

- Logic: Advanced lifters lift heavier → more damage → need more sleep

Goal-Based Adjustment

- Logic: Strength/power training depletes CNS more → needs neural recovery

Calculate Total Recommendation

- Logic: Strength/power training depletes CNS more → needs neural recovery

SLEEP RECOMMENDATIONS

```text
Base Sleep:          7.5 hours
Volume Factor:       5 days × 0.2 = +1.0 hour
Experience Factor:   Intermediate = +0.3 hours
Goal Factor:         Weight Loss = 0 hours
                     ──────────────────────
TOTAL RECOMMENDED:   7.5 + 1.0 + 0.3 + 0 = 8.8 hours/night
```

```text
SLEEP RECOVERY RECOMMENDATIONS

🎯 Target Sleep: 8.8 hours/night
    (adjusted for 5-day training + intermediate recovery needs)

💡 Tips:
1. Maintain consistent sleep schedule (sleep at same time daily)
2. Avoid screens 1 hour before bed
3. Keep room cool (60-67°F / 15-19°C)
4. Stay hydrated throughout the day
5. If high volume: Prioritize sleep for recovery
6. Consider naps: 20-30 min power naps mid-day if needed

⚠️ Special Notes:
   └─ Heavy training volume (5 days) requires prioritizing sleep
   └─ Your arm splits need especially good recovery
```

Future Improvements

- The current system is production-ready with 86.81% accuracy, but we've identified strategic improvements that would further enhance personalization, safety, and user engagement. Below are prioritized enhancements based on research and user feedback patterns.

1-UNSUPERVISED PERSONALIZATION (more User-Driven Inputs)
2-Movement Family Diversity Filtering
3-INJURY HANDLING IMPROVEMENTS using NLP
4-CONTENT-BASED FILTERING IMPROVEMENTS (dynamic threshold adjustments)
5- USER EXPERIENCE ENHANCEMENTS 'feedback loops'
6-Exercise Explanation & Reasoning
7-Collaberitive filtering add + Cold-Start Problem Solution
8-Nutrition meals recommendations
9-Warmups stretches dynamic (currently predefined)

Related work

611noorsaeed/**Diet-and-Workout-...**

Diet and Workout Recommendation system using openai

1 Contributor · 1 Issue · 22 Stars · 8 Forks

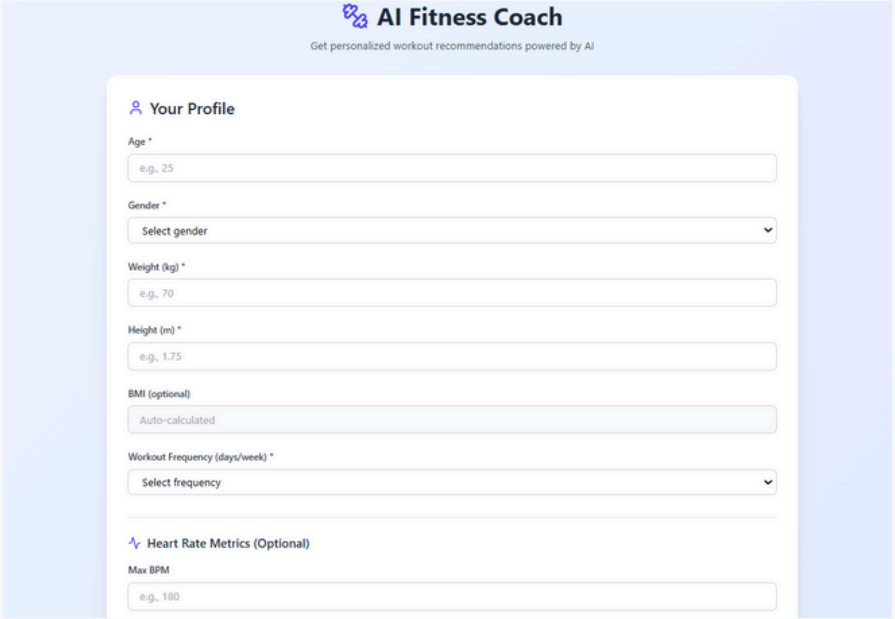
611noorsaeed/Diet-and-Workout-Recommendation-system-using-openai: Diet and Workout Recommendati...

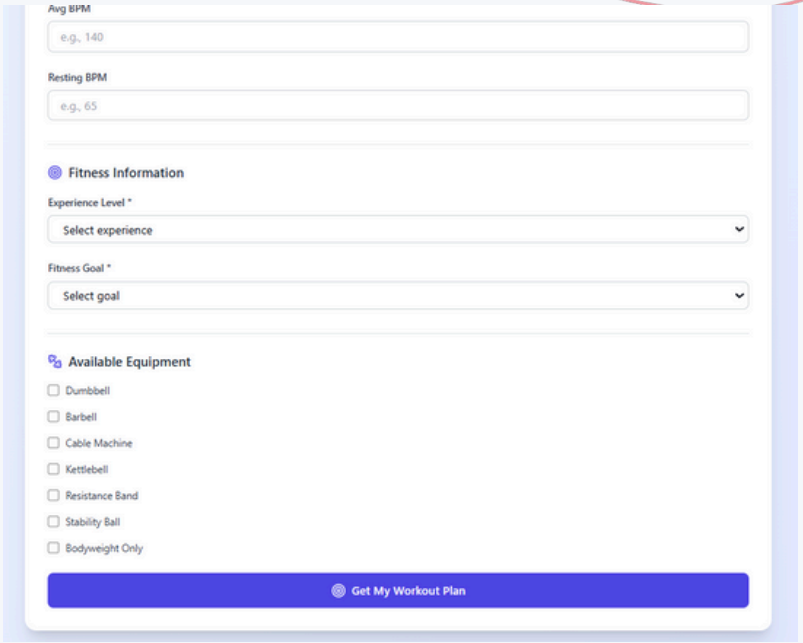
Diet and Workout Recommendation system using openai - 611noorsaeed/Diet-and-Workout-Recommendation-system-using-openai

GitHub

```
6]: workout_names

6]: ['1. Walking',
     '2. Swimming',
     '3. Yoga',
     '4. Pilates',
     '5. Strength Training',
     '6. Cycling']
```

Deployment


AI Fitness Coach
Get personalized workout recommendations powered by AI
Your Profile
Age *
e.g., 25
Gender *
Select gender
Weight (kg) *
e.g., 70
Height (m) *
e.g., 1.75
BMI (optional)
Auto-calculated
Workout Frequency (days/week) *
Select frequency
Heart Rate Metrics (Optional)
Max BPM
e.g., 180

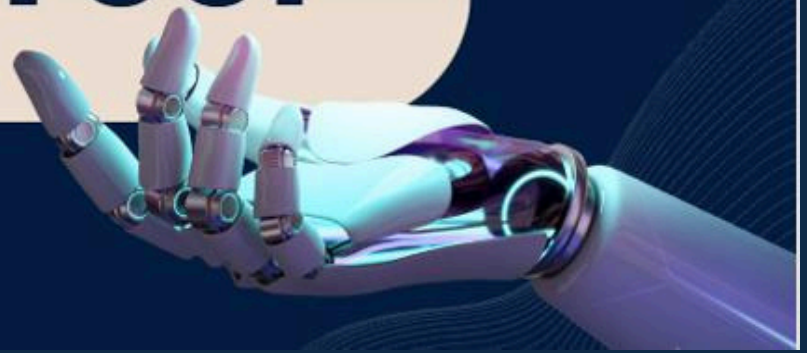

Avg BPM
e.g., 140
Resting BPM
e.g., 65
Fitness Information
Experience Level *
Select experience
Fitness Goal *
Select goal
Available Equipment
Dumbbell
Barbell
Cable Machine
Kettlebell
Resistance Band
Stability Ball
Bodyweight Only
Get My Workout Plan

TOOLS

رواد مصر الرقمية

وزارة الاتصالات وتكنولوجيا المعلومات

THANK YOU!



Youth co.