وزارة الاتصــــالات
وتكنولوجيا المعلومات

رواد مصر الرقمية

Recommendation System for Fitness and Nutrition

Supervised by **Eng. Mahmoud Talaat**

Youth co.

Done by
Team members

Mariam Mahmoud

Hanya Wael

Eman Mostafa

Nour Mohamed

TABLE OF CONTENTS

01

Introduction

Introduction

The global fitness industry is experiencing remarkable growth, driven by increased 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 and nutrition plans** 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.

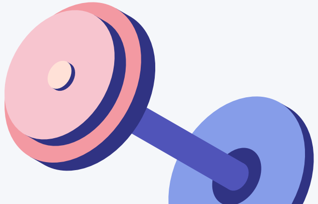
02

Problem statement

Current Challenges in Achieving Personalized Fitness

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

- 🏋️ **Generic workout plans** fail to consider users' individual goals, physical abilities, or health conditions — leading to ineffective results and loss of motivation.
- 📉 **Lack of professional guidance** makes 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 the limitations of traditional, one-size-fits-all workout approaches, we have developed an **AI-based Fitness Recommendation System** that provides **personalized training and nutrition guidance** tailored to each user’s unique needs.

🏠 **Personalized Fitness Guidance:**

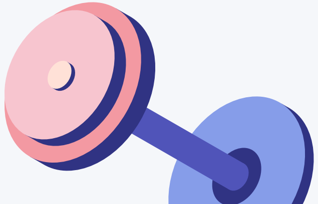
The system allows users to input key personal data — such as age, weight, fitness goals, activity level, and health conditions — to receive customized workout and nutrition plans suited to their profile.

🤖 **AI-Powered Recommendation Engine:**

Using artificial intelligence and data analytics, the system analyzes user data to recommend exercises, meal plans, and progression schedules. It dynamically updates recommendations as users improve or their goals change.

💪 **Goal:**

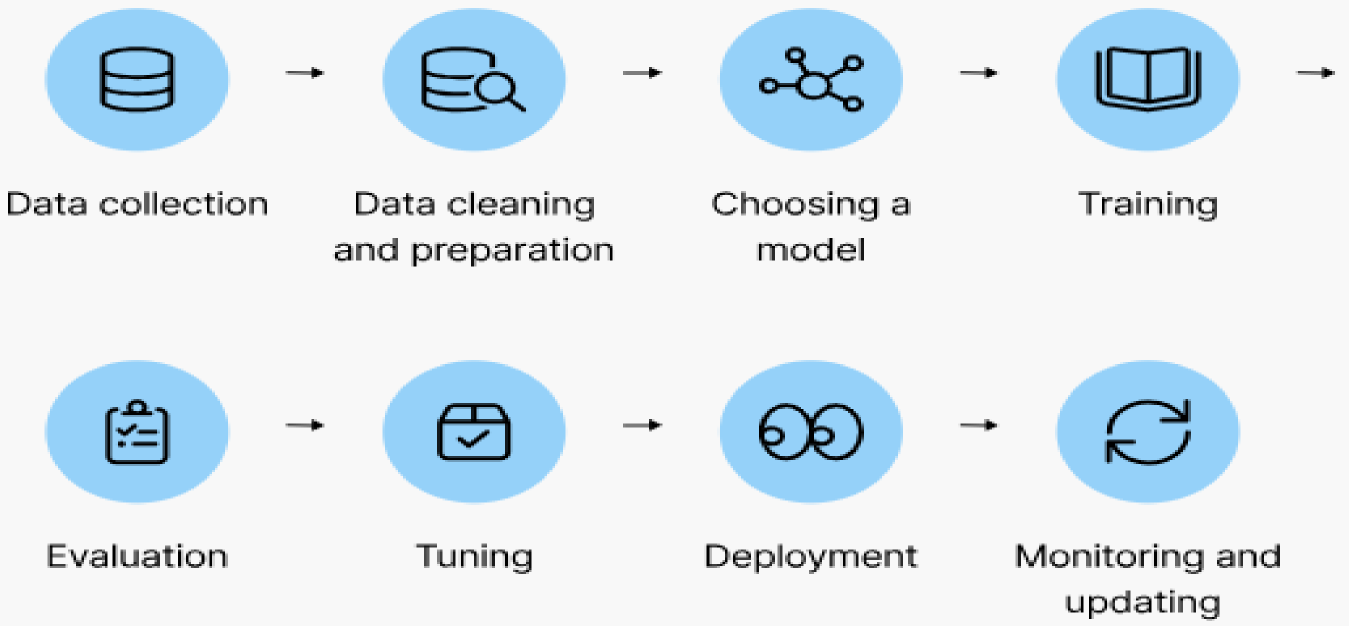
To make fitness training **smarter, safer, and more effective** by providing a **personalized, data-driven solution** that enhances motivation, consistency, and long-term health outcomes.

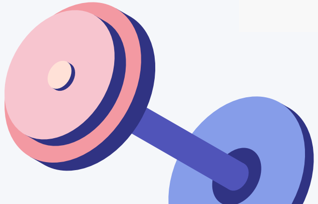


04

Methodology

Methodology





Data collection

Source:

Explanation features of data:

- 🧍 ♂ **Gender:** Identifies whether the user is male, female, or other.
- 🎂 **Age:** Represents the user’s age in years, used to tailor workout intensity and type.
- ⚖️ **BMI (Body Mass Index):** Indicates body composition and helps assess appropriate fitness plans.
- ❤️ **Heart Rate / Health Conditions:** Reflects user health status to avoid unsafe workout recommendations.
- 🚬 **Lifestyle Habits:** Includes indicators such as smoking or activity levels, which influence fitness recommendations.
- 🏃 **Activity Level:** Represents how active a person currently is (e.g., sedentary, moderate, active).
- 🍎 **Diet Type:** Helps in suggesting nutrition plans compatible with the user’s fitness goals.
- 🎯 **Goal Type:** Specifies user’s target (e.g., weight loss, muscle gain, endurance).

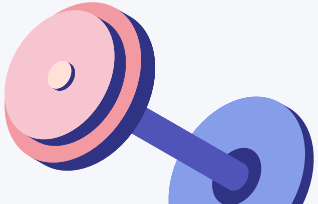
Data cleaning

First, we **removed the duplicated rows** from the data **to ensure each row contains unique information**, which improves the quality of the data analysis.
Then, we displayed the data again to **verify the changes** and ensure the duplicated values were successfully removed.
Finally, we **checked** for **missing (NAN) values** in the columns to identify any empty values that might affect the analysis, allowing us to address them appropriately.

```
users=users.drop_duplicates()
users
```

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

```
# Check missing values per column
print(exercises_df.isna().sum())
```



```
# Show number of duplicates
print(exercises_df.duplicated().sum())
```

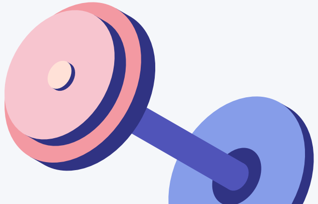
Data analysis

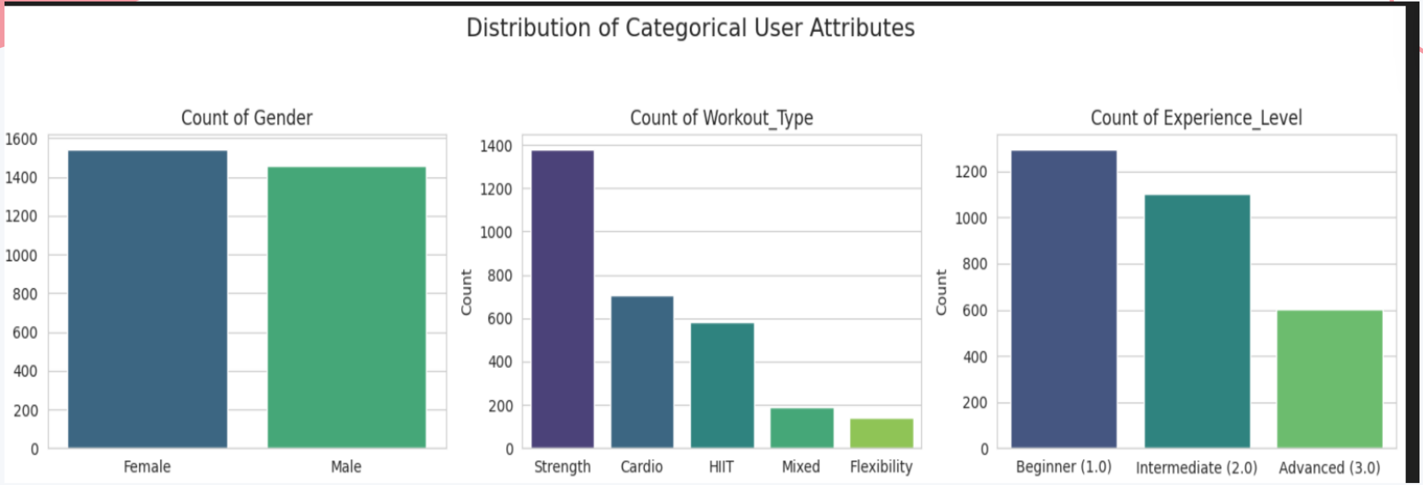
In data analysis, we use key commands to understand the dataset before training. **users.info()** provides basic information about the columns, such as data types, non-null values, and overall structure. **users.nunique()** shows the number of unique values in each column, helping us identify whether 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.

```
# Show dataset information
users.info()
```

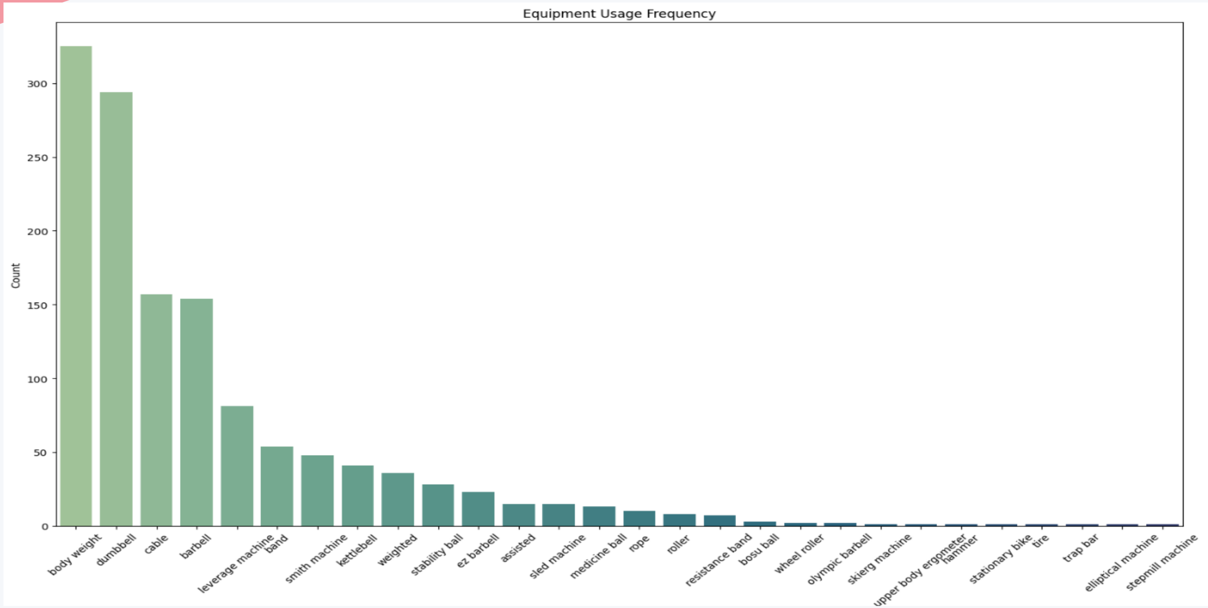
```
users.nunique()
```

```
# Get summary statistics
stat_df = users.describe()
```

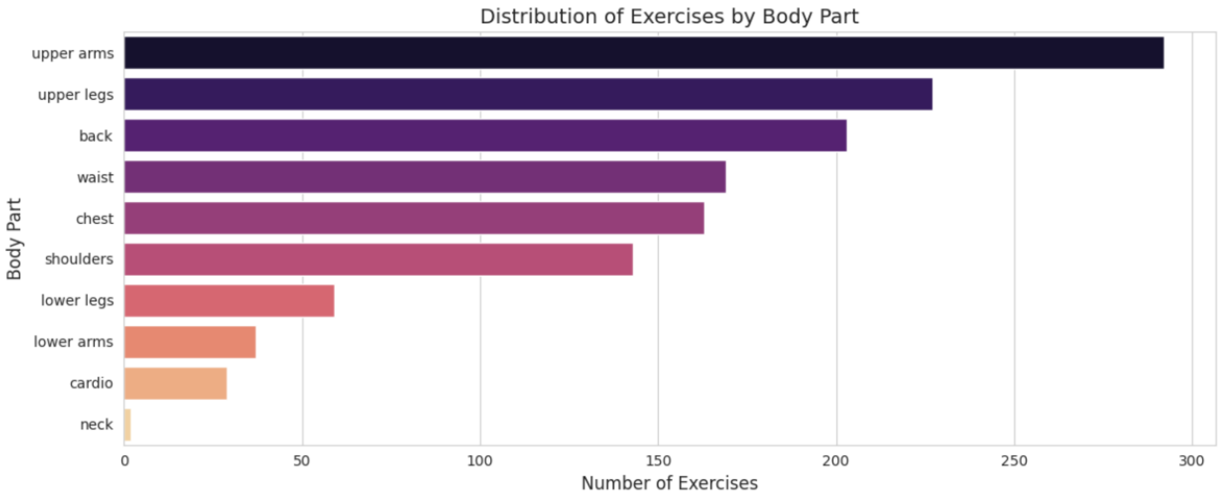




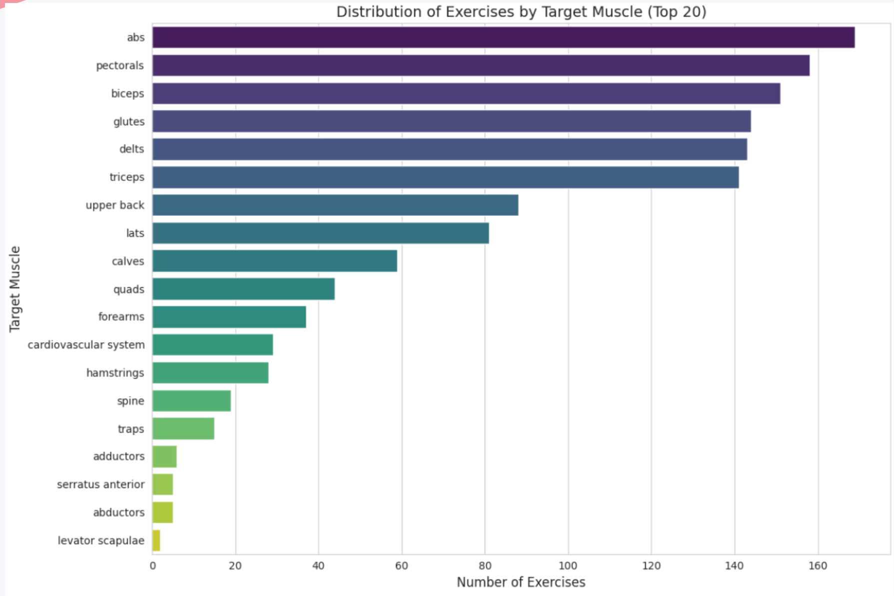
•**Gender Distribution (Count of Gender):** The dataset is fairly balanced, with a slight majority of users identifying as **Female** (approximately 1,550) compared to **Male** (approximately 1,450).
•**Workout Preference (Count of Workout_Type):** There is a strong preference for **Strength** training, which is the most popular type by a large margin (approx. 1,380 users). This is followed by **Cardio** (approx. 700) and **HIIT** (approx. 580). "Mixed" and "Flexibility" are the least common workout types.
•**Experience Level (Count of Experience_Level):** The data is skewed towards less experienced users. **Beginners (1.0)** make up the largest group (approx. 1,250), followed closely by **Intermediates (2.0)** (approx. 1,100). **Advanced (3.0)** users are the smallest cohort, representing about half the number of intermediates.



•The dataset is dominated by common, accessible equipment. **Body weight** and **dumbbell** exercises are by far the most frequent, followed by standard gym equipment like **cable** and **barbell**. There is a sharp drop-off after these top four, with most other equipment types being far less common.



•**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.



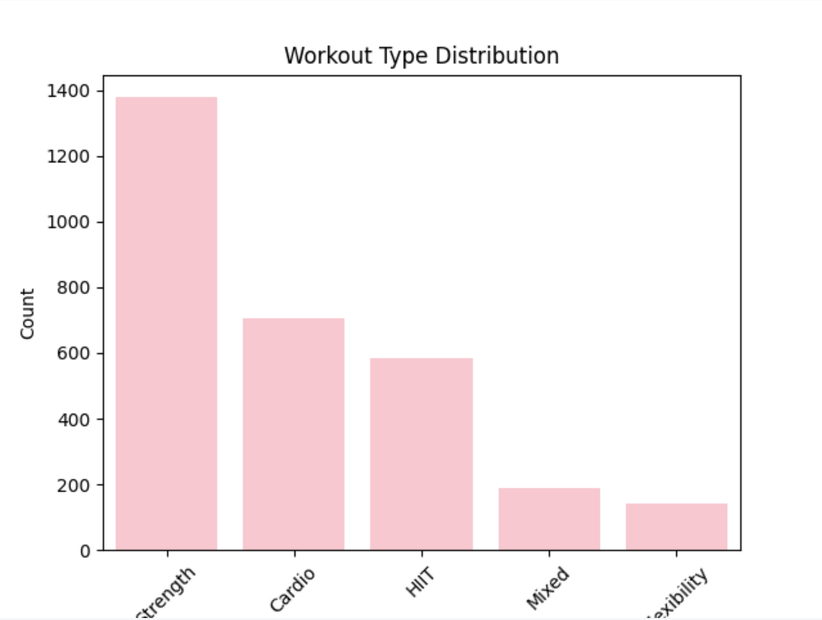
- **Core & "Push" Muscles Dominate:** The dataset has the highest variety of exercises for **abs** (approx. 165) and **pectorals** (chest, approx. 155). This is followed closely by other popular "push" and "pull" day muscles: **biceps**, **glutes**, **delts** (shoulders), and **triceps** (all between 140-150 exercises).
- **Significant Leg Imbalance:** There is a major discrepancy in leg exercises. While **glutes** are very well-represented, **quads** (approx. 45) and especially **hamstrings** (approx. 25) have significantly fewer dedicated exercises in the top 20.
- **Back Muscle Split:** The dataset clearly distinguishes between different back muscles. **Upper back** (approx. 95) and **lats** (approx. 85) have a good number of exercises, but **traps** and **spine**-focused exercises are far less common.
- **Cardio & Forearms are Low: Cardiovascular system** (approx. 30) and **forearms** (approx. 35) are in the lower tier, confirming the dataset's primary focus on resistance training for major muscle groups.
- **Niche/Accessory Muscles:** Muscles like adductors, serratus anterior, abductors, and levator scapulae have very few exercises, making up the bottom of the list.

Calories Burned by Workout Type and Gender

Across most workout types (Strength, Cardio, HIIT), males and females burn a similar number of calories.
In **Mixed workouts**, males burn slightly more calories, while in **Flexibility workouts**, both genders burn the least calories overall.
This indicates that **workout type is a stronger factor than gender** in calorie expenditure.

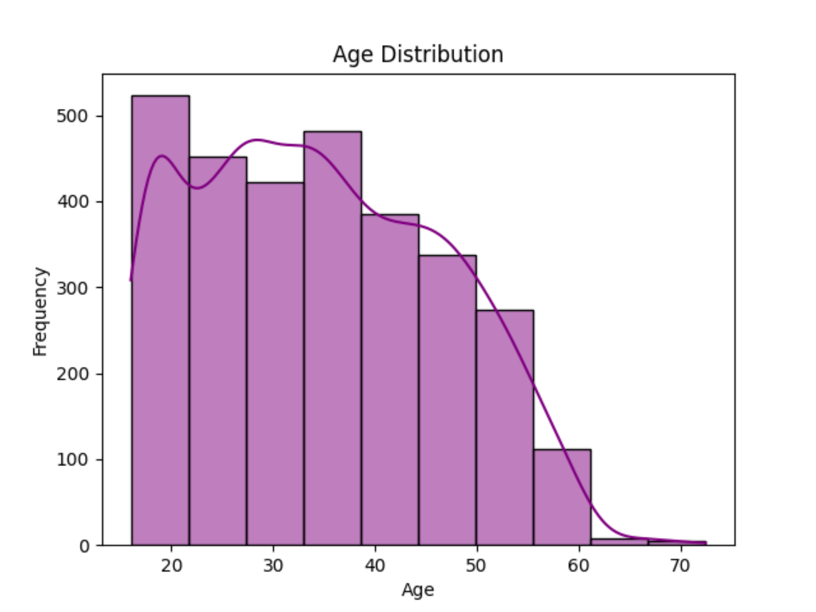
Workout Type Distribution

Strength training is the most common workout type, followed by **Cardio** and **HIIT**.
Mixed and **Flexibility workouts** are the least common, suggesting users prefer higher-intensity or strength-focused routines



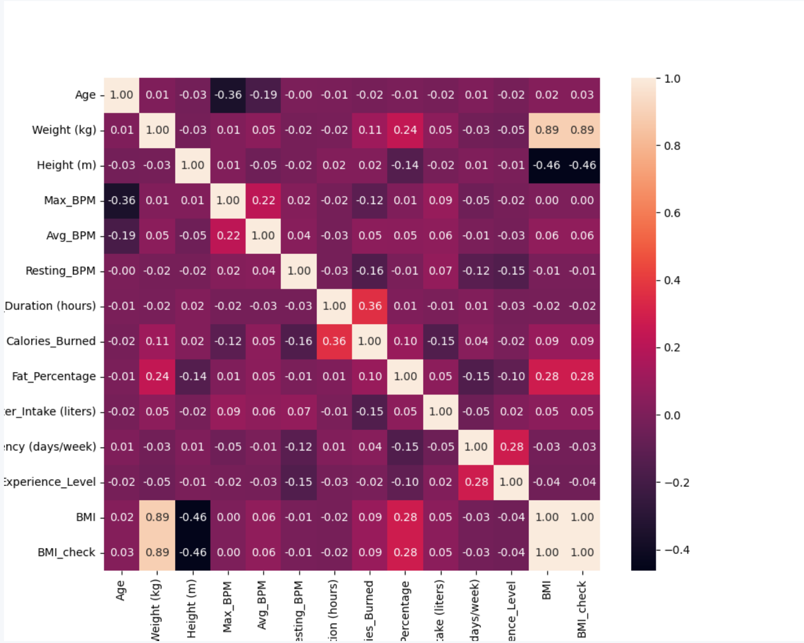
Age Distribution of Users

Most users fall between **18–40 years old**, showing that younger adults are the primary audience.
Participation decreases significantly after age **50**, suggesting lower adoption among older groups.



Correlation Analysis

Longer workout sessions lead to more calories burned(**Calories Burned & Duration (0.83)**)
More experienced individuals tend to work out more consistently throughout the week(**Experience Level & Efficiency (0.78)**)
Taller individuals may be less likely to trigger BMI alerts, possibly due to how BMI scales with height(**BMI Check & Height (-0.28)**)
Higher water intake might be associated with lower fat percentage, hinting at healthier habits (**BMI Check & Height (-0.28)**).



Data preprocessing

Cleaning the Exercises Dataset

Combined Scattered Columns:

Merged multiple “secondaryMuscles” columns into a single list for each exercise.

Joined all “instructions” columns into one clear, readable text field.

Removed Irrelevant Columns:

Dropped unnecessary fields such as *gifUrl* and redundant instruction or muscle columns.

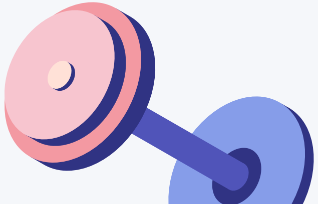
Enhanced Dataset Structure

Produced a cleaner, more compact dataset with:

Unified columns for easier processing.

Consistent formatting ready for feature extraction and recommendation modeling.

```python
print("\n" + "=" * 60)
print("1️⃣ DATA LOADING AND PREPROCESSING")
print("=" * 60)

# Load datasets
users = pd.read_csv('users.csv')
exercises_raw = pd.read_csv('excercises.csv')

print(f"📊 User dataset: {users.shape[0]:,} users with {users.shape[1]} features")
print(f"🏃 Exercise dataset: {exercises_raw.shape[0]:,} exercises with {exercises_raw.shape[1]} features")

def clean_exercises_dataset(df):
    """Clean and process the exercise dataset with enhanced features"""
    clean_df = df.copy()

    # Combine secondary muscles
    secondary_cols = [col for col in df.columns if 'secondaryMuscles' in col]
    clean_df['secondary_muscles'] = df[secondary_cols].apply(
        lambda row: [x for x in row if pd.notna(x)], axis=1
    )

    # Combine instructions
    instruction_cols = [col for col in df.columns if 'instructions' in col]
    clean_df['instructions'] = df[instruction_cols].apply(
        lambda row: ' '.join([x for x in row if pd.notna(x)]), axis=1
    )

    # Drop original scattered columns
    cols_to_drop = secondary_cols + instruction_cols + ['gifUrl']
    clean_df = clean_df.drop(columns=cols_to_drop, errors='ignore')

    # Enhanced difficulty mapping
```



Data preprocessing

Enriching Exercise Data for Smarter Recommendations

🧠 1. Difficulty Level Mapping

Assigned a **numerical difficulty score (1–3)** based on exercise equipment type.
Example: Bodyweight = 1 (Easy), Dumbbell = 2 (Moderate), Barbell = 3 (Hard).
Helps the model recommend workouts appropriate to each user's fitness level.

🏋️ 2. Exercise Type Classification

Grouped exercises into broad categories such as:
Strength, **Cardio**, **Core**, and **Flexibility**
Simplifies model training and recommendation filtering by body part.

🏠 3. Accessibility Score

Rated each exercise on a scale of **1 (Gym Only)** → **5 (Home Friendly)**
Enables the system to tailor recommendations based on equipment availability.

📏 4. Complexity Measurement

Calculated a **complexity score (1–5)** based on instruction length — longer instructions often indicate higher skill or coordination requirements.

📊 5. Dataset Summary & Validation

Verified the diversity of features such as **body parts**, **equipment types**, and **target muscles**.
Ensured a balanced dataset for generating accurate and inclusive fitness recommendations.

```python
    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:")
```

Data preprocessing

The raw excercises.csv data was fragmented, with instructions and secondary muscles spread across multiple columns (e.g., instructions/0, instructions/1, secondaryMuscles/0, etc.).

To fix this, we:

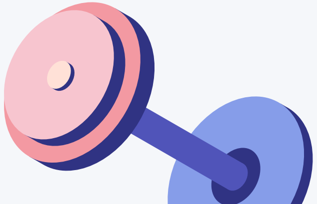
Consolidated Muscles: Combined all secondaryMuscles columns into a single, clean secondary_muscles text field.

Combined Instructions: Merged all instructions columns into one instructions_full field for complete details.

Engineered instructions_short: Created a new column by extracting only the first 3 steps to use as a quick preview in the final plan.

Cleaned Dataset: Kept only 8 relevant columns (like name, target, equipment, and our new fields) to prepare the data for the recommendation engine.

```python
# 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])
```



05

Machine Learning Results

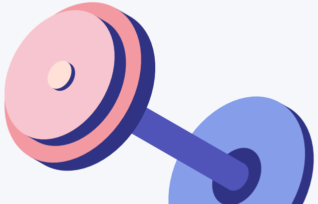
Models Training

There are 7 models that was tried in this project, which are:

1. Random Forest
2. Decision Tree
3. neural networks
4. SVM
5. collaborative filtering+ content based
6. XGBOOST
7. Clustering

stable approaches in our project :

1. SVM
2. collaborative filtering+ content based
3. XGBOOST

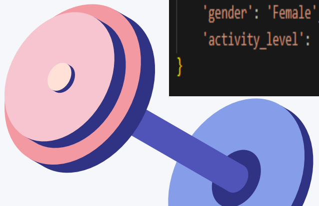


MODEL I/O

```python
# ========================================
# 🔥 CUSTOMIZE YOUR PLAN HERE 🔥
# ========================================

# BASIC SETTINGS
NUM_DAYS = 4      # 3-7 for Strength, 2-4 for Flexibility
EXPERIENCE_LEVEL = 2       # 1=Beginner, 2=Intermediate, 3=Advanced
WORKOUT_TYPE = 'Strength'   # 'Strength', 'Flexibility', 'Cardio', 'HIIT'

# GOAL (affects sets, reps, rest, exercise selection)
GOAL = 'fat loss'       # 'hypertrophy', 'strength', 'fat loss', 'endurance', 'power' only applied to strength

# EQUIPMENT AVAILABLE
#EQUIPMENT_AVAILABLE = ['dumbbell', 'barbell', 'body weight', 'cable', 'band']
EQUIPMENT_AVAILABLE = None  # Uncomment for all equipment

# INJURIES/RESTRICTIONS (prevents dangerous exercises)
INJURIES = ['shoulder injury']  # Options: 'shoulder injury', 'lower back pain', 'knee injury',
                                # 'elbow pain', 'wrist pain', 'hip injury', 'neck pain', 'ankle injury'

# USER PROFILE FOR MACROS & SLEEP
USER_PROFILE = {
    'age': 22,
    'weight_kg': 75,
    'height_cm': 170,
    'gender': 'Female',
    'activity_level': 'active'  # 'sedentary', 'light', 'moderate', 'active', 'very_active'
}
```

```
🔄 Generating your professional workout plan...

==========================================================================================
                        🏋 PERSONALIZED WORKOUT PLAN 🏋
==========================================================================================

📋 PROGRAM DETAILS:
   Days: 3/week | Level: Intermediate | Goal: FAT LOSS
   Type: Strength | Equipment: All available
   ⚠ Injuries: shoulder injury

==========================================================================================
🗓 DAY 1: Chest + Shoulders + Triceps
==========================================================================================

🔥 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
...
  • Aim for 8.4 hours of sleep per night
  • Maintain consistent sleep/wake times
  • Avoid screens 1 hour before bed
  • Keep room cool (60-67°F / 15-19°C)
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...



imbalanced Data Handling

1. Problem Identification

During data exploration, class distribution among fitness categories (e.g., workout types or nutrition levels) was **uneven** — some categories had far more samples than others.

This imbalance can bias the model toward predicting the **majority class**, reducing fairness and accuracy for underrepresented user groups.

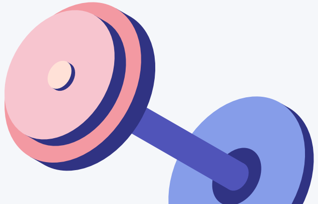
Techniques Used to Fix Imbalance

✅ **SMOTE (Synthetic Minority Oversampling Technique)**

Generates **synthetic samples** for minority classes instead of duplicating existing data.

Creates new samples by **interpolating between nearest neighbors**, helping the model learn smoother decision boundaries.

Ensures that all fitness categories are **equally represented** in the training data.



06

Conclusion

Conclusion

1. Summary of the Project

Developed an **AI-driven Fitness Recommendation System** that delivers **personalized workout and nutrition plans** based on user data and fitness goals.

Implemented comprehensive **data preprocessing, feature engineering, and machine learning models** to ensure accurate and adaptive recommendations.

🤖 **2. Model Performance**

Among several tested algorithms, **Support Vector Machine (SVM)** achieved the best accuracy and generalization performance.

The system provides reliable and tailored fitness suggestions to match individual needs.

🏋️ **3. Impact**

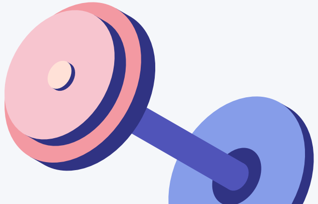
Promotes **smart, data-based fitness planning** that improves user engagement, safety, and results.

Bridges the gap between **generic fitness apps** and **personalized training programs**.

🚀 **4. Future Work**

Integrate **real-time progress tracking** and **wearable device data**.

Add **nutrition recommendations** and **AI-powered chat support** for holistic health management.



TOOLS

رواد مصر الرقمية

وزارة الاتصالات وتكنولوجيا المعلومات

THANK YOU!

YOUth co.