Tunisian Republic
Ministry of Higher Education and Scientific Research
Private Higher School of Information Technology and Management in Nabeul

Final Study Project Report

Final Study Report submitted to obtain the

National Engineering Diploma in Computer Engineering
Spécialization: Business Intelligence

Prepared by

Akram Nemri & Oussama Zid

Machine Learning-Powered Fitness Prediction and Workout Recommendation System

---

Academic Advisor: M Ahmed Ben Taleb **Professional Advisor:** M Foulen

Final Study Report done at

Host Company Placeholder

Academic Year 2024-2025

Deposit Agreement

Deposit Agreement for Final Study Report

This document certifies that the Final Study Report has been deposited at the hosting institution according to the submission rules and regulations. The undersigned parties acknowledge the deposit and accept the terms of submission and archiving.

Submitted by: Akram Nemri & Oussama Zid
Institution:
Private Higher School of Information Technology and Management in Nabeul
Date:

Signature: ____

Dedication

To our families, for their unwavering sacrifice and support, a testament to our infinite gratitude and deep affection.

To all those dear to us...

Acknowledgment

We would like to express our sincere gratitude to all those who contributed to the successful completion of this project.
We are deeply thankful to our academic advisor, **M Ahmed Ben Taleb**, for his valuable guidance, availability, and constructive criticism throughout this work. His expertise was instrumental in shaping the technical direction of this project.
We also extend our thanks to the faculty members of
Private Higher School of Information Technology and Management in Nabeul for providing the foundational knowledge that made this work possible.
Finally, a special acknowledgment to our friends and family for their continuous emotional support and encouragement.

Abstract

Machine Learning-Powered Fitness Prediction and Workout Recommendation System

Abstract

This project focuses on developing a robust Machine Learning (ML) system to enhance personal fitness tracking and guidance. Utilizing a synthetic dataset of gym members' exercise activities, the system addresses three key challenges: predicting the calories burned during a session, forecasting future body weight based on exercise data, and recommending optimal workout types aligned with a user's fitness goals (lose weight, gain muscle, maintain). We implement and compare multiple regression and classification models, specifically **Linear Regression**, **Support Vector Machines** (Support Vector Machines (SVM)), and **Gradient Boosting** (Gradient Boosting (GB)), within a cohesive **Scikit-learn Pipeline** architecture. The calorie and weight prediction tasks employ regression models, while the workout recommendation is solved using a classification approach, with GB and SVM classifiers showing strong performance. The final deployed models provide a powerful backend for a personalized fitness application, offering data-driven insights to help users achieve their health objectives more efficiently.

Keywords — Machine Learning, Fitness Prediction, Calories, Weight Forecasting, Workout Recommendation, Gradient Boosting, SVM, Scikit-learn.

Résumé (Executive Summary)

Machine Learning-Powered Fitness Prediction and Workout Recommendation System

This project focuses on developing a robust machine-learning system to improve personal fitness tracking and guidance. Using a synthetic dataset based on gym members' workout activities, the system tackles three key challenges: predicting the calories burned during a session, forecasting future body weight based on exercise data, and recommending optimal types of training aligned with a user's fitness goals (weight loss, muscle gain, maintenance). We implement and compare several regression and classification models, including Linear Regression, Support Vector Machines (SVM), and Gradient Boosting (GB), all within a consistent Scikit-learn Pipeline architecture. Calorie and weight predictions use regression models, while workout recommendation is approached as a classification task, with GB and SVM classifiers showing strong performance.

The final deployed models form a powerful backend for a personalized fitness application, delivering data-driven insights to help users reach their health goals more effectively.

Contents

List of Figures

List of Tables

Acronyms

GB Gradient Boosting. i, 6–11, 13

LogR Logistic Regression. 7, 10

LR Linear Regression. 6, 7

MAE Mean Absolute Error. 8

ML Machine Learning. i, v, 1, 11, 13

RMSE Root Mean Square Error. 8, 9

SVM Support Vector Machines. i, iii, iv, 3, 6–9

Chapter 1

Introduction

The intersection of technology and fitness has led to the rise of intelligent personal health applications. These tools, powered by ML algorithms, transition from simple data logging to proactive, personalized guidance. The core objective of this project is to develop a comprehensive ML system capable of addressing three critical aspects of personal fitness: predicting workout outcomes, forecasting future body metrics, and providing goal-oriented exercise recommendations.

1.1 Problem Definition

Individuals often struggle to maintain consistency and maximize the effectiveness of their fitness regimes due to a lack of precise, personalized feedback. The project addresses this by tackling three distinct prediction and recommendation challenges:

1. **Calories Prediction:** Accurately estimating the calories burned during a workout session is fundamental for managing energy balance (intake vs. expenditure).
2. **Weight Prediction:** Forecasting a user's weight over a period of time, based on their current metrics and exercise habits, provides essential motivation and long-term planning capability.
3. **Workout Recommendation:** Providing a specific workout type (e.g., Cardio, Strength, HIIT) that aligns with a user's primary fitness goal (lose weight, gain muscle, maintain).

1.2 Project Approach

To solve these problems, we utilize a supervised learning approach using the `gym_members_exercise_tracking_synthetic_data.csv` dataset. The solution involves building a robust ensemble of models, including Gradient Boosting and Support Vector Machines, to ensure high predictive accuracy and reliable recommendations.

1.3 Report Structure

The remainder of this report is organized as follows: Chapter 2 details the dataset, including its features and the preprocessing steps applied. Chapter 3 covers the methodology, presenting the specific Machine Learning models and pipelines developed for each of the three tasks. Chapter 4 presents the results and model comparisons. Finally, Chapter 5 presents the discussion, and the report concludes with Chapter 5.4 on the overall conclusion, limitations, and future perspectives.

Chapter 2

Dataset and Preprocessing

2.1 Dataset Overview

The project relies on a synthetic dataset named `gym_members_exercise_tracking_synthetic_data.csv`. This dataset is designed to simulate tracking data from gym members, providing a rich collection of physiological, behavioral, and workout-specific features necessary for our predictive models.

2.1.1 Key Features

The dataset includes both quantitative and qualitative features. Critical quantitative features used across our models include:

- Age, Weight (kg), Height (m), BMI, Fat_Percentage
- Avg_BPM, Resting_BPM
- Session_Duration (hours), Calories_Burned (Target variable for prediction)

The primary qualitative feature is `Workout_Type`, which is the target for the recommendation model. The dataset initially contains 1800 entries with varying levels of missing data (as shown by the Non-Null Counts).

2.1.2 Exploratory Data Analysis (EDA)

During the EDA phase, several key relationships were observed:

- **Correlation Matrix:** A heatmap (Figure 2.1) revealed strong correlations between `Calories_Burned` and features like `Session_Duration (hours)` and `Avg_BPM`.
- **Distribution of Target:** The distribution of `Calories_Burned` (Figure 2.2) was analyzed to ensure sufficient variance for regression tasks.

2.2 Data Preprocessing

Consistent data cleaning was applied across all models to ensure robustness and comparability. The preprocessing steps, as evident in the provided Python scripts, include:

2.2.1 Data Cleaning

1. **Numeric Cleaning:** Tab characters (`\t`) and excessive whitespace were removed from all numeric columns, and the columns were explicitly converted to a numeric type using `pd.to_numeric` with `errors='coerce'` to handle non-convertible values as missing data (`NaN`).
2. **Categorical Cleaning:** The `Workout_Type` column was stripped of special characters (tabs, newlines) and empty strings/literal 'nan' strings were converted to `NaN`.

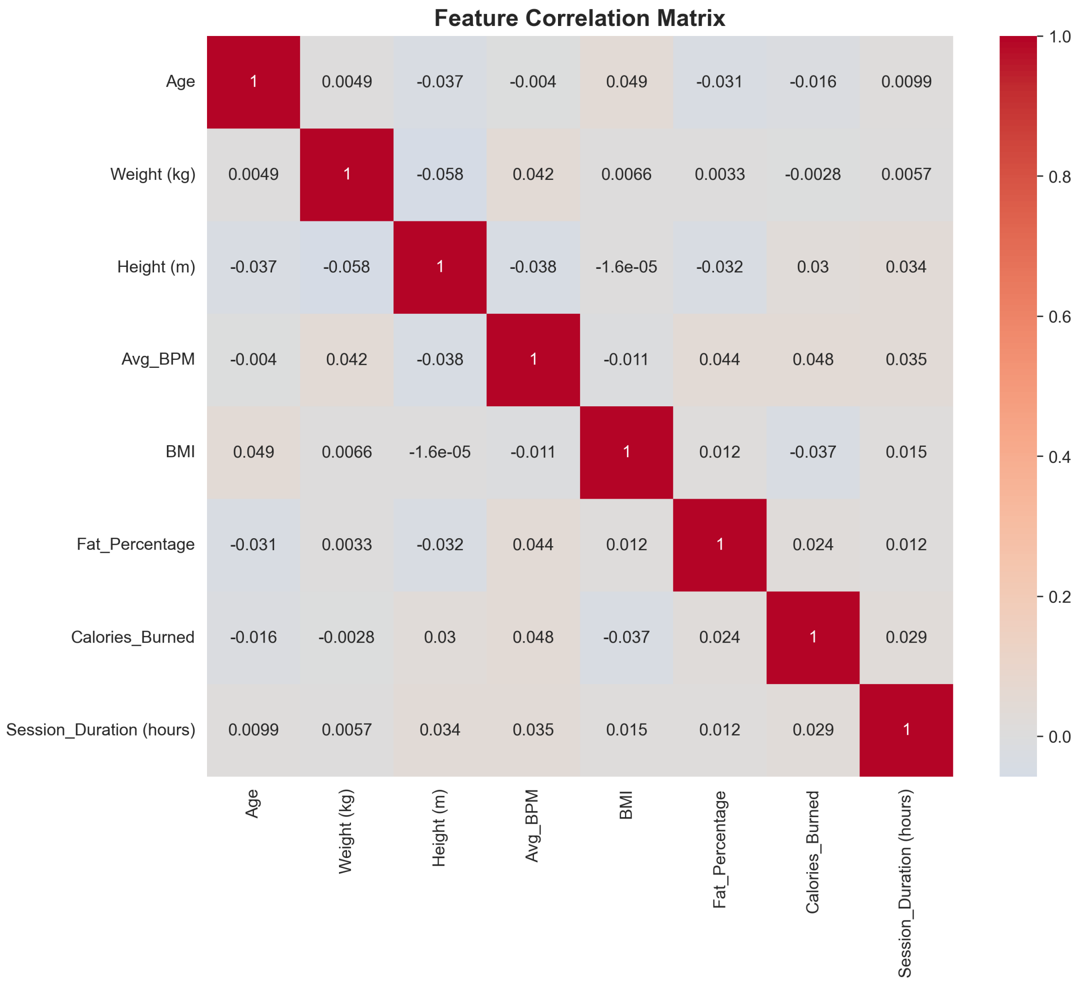

Figure 2.1: Correlation Heatmap of Key Numerical Features.

2.2.2 Missing Value Handling

Rows with missing values (`NaN`) in critical features required for model training (e.g., `Age`, `Weight (kg)`, `Calories_Burned`, `Workout_Type`) were dropped using `df.dropna(subset=critical_cols)`. This reduced the dataset size to approximately 1600 rows.

2.2.3 Feature Engineering and Scaling

1. **Goal Feature Creation:** For the workout recommendation task, a primary feature, `goal`, was engineered based on the `BMI` metric:

   - `BMI < 18.5` → `gain_muscle`
   - `BMI > 25` → `lose_weight`
   - Otherwise → `maintain`

   This goal feature was then Label Encoded for use in classification models.

2. **Scaling for SVM:** For the SVM models (both regression and classification), the numerical features were scaled using `StandardScaler` to ensure the algorithm's performance is not hindered by varying feature magnitudes.

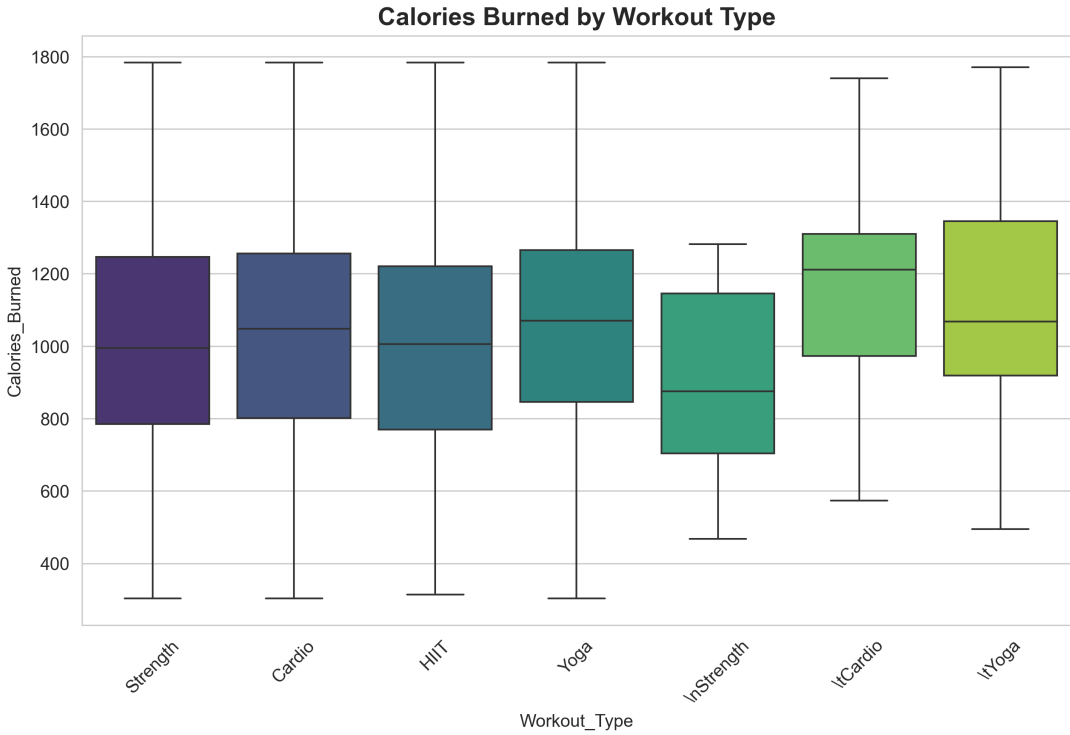

Figure 2.2: Distribution of the Target Variable: Calories Burned.

3. **One-Hot Encoding:** Categorical features (`Workout_Type`) were transformed using `OneHotEncoder` as part of the Scikit-learn Pipeline for all models.

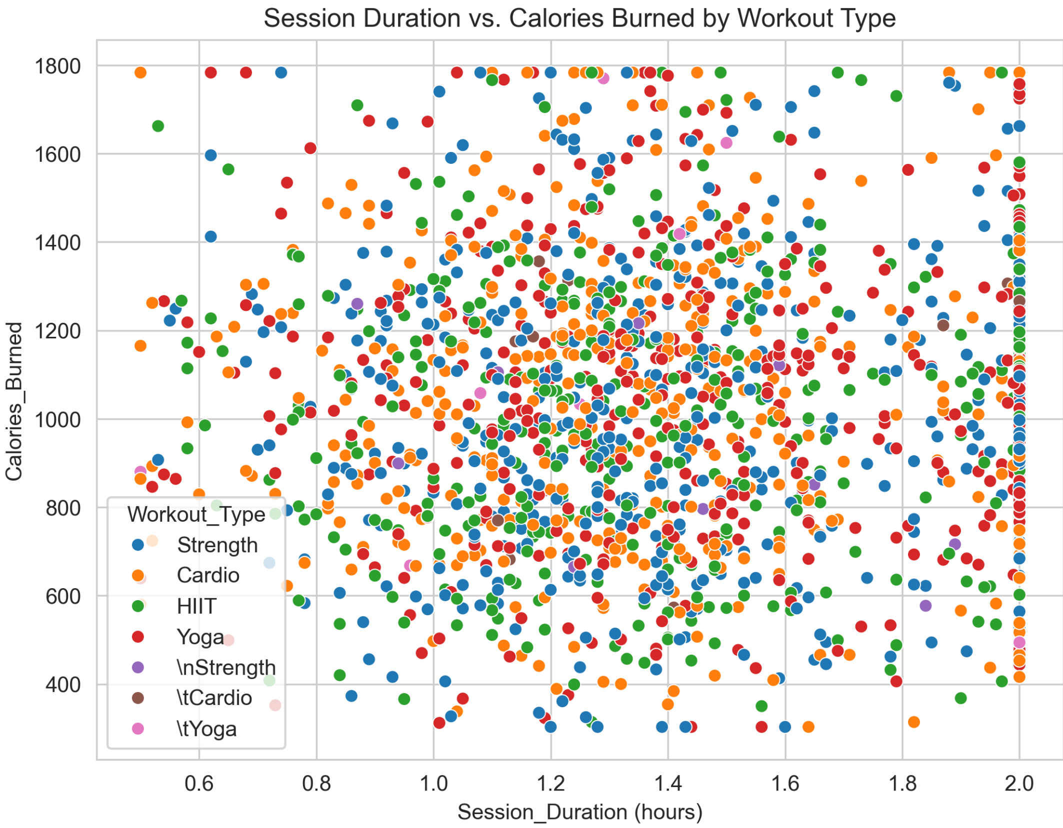

Figure 2.3: Scatter Plot showing the positive relationship between Session Duration and Calories Burned.

Chapter 3

Methodology (Models Used and Applications)

Our system is structured around three distinct Machine Learning tasks: two regression problems for prediction (Calories and Weight) and one classification problem for recommendation (Workout Type). Each task explores multiple algorithms to determine the optimal approach in terms of accuracy and robustness.

3.1 Calories Prediction Model

The objective of this model is to estimate the `Calories_Burned` based on session metrics and user physiology. This is a classic regression problem.

3.1.1 Features and Target

- **Target:** `Calories_Burned`
- **Features:** `Age`, `Weight (kg)`, `Height (m)`, `Avg_BPM`, `Resting_BPM`, `Session_Duration (hours)`, `Fat_Percentage`, `Workout_Frequency (days/week)`, `Experience_Level`, `BMI`, `Workout_Type`.

3.1.2 Model Pipelines

We evaluated three different regression algorithms: Linear Regression (LR), GB, and SVM Regression. The data transformation for all models was encapsulated within a `Pipeline` for consistency.

Linear Regression (Benchmark Model)

The LR model (script `train_calories_model.py`) served as a simple, interpretable benchmark. The pipeline primarily consisted of `OneHotEncoder` for the categorical `Workout_Type` followed by the `LinearRegression` estimator.

Gradient Boosting Regressor

The GB Regressor (script `train_calories_model_gradient_boosting.py`) was chosen for its proven ability to handle complex non-linear relationships and interactions between features.

- **Pipeline:** `ColumnTransformer` (only `OneHotEncoder` on `Workout_Type`) $\rightarrow$ `GradientBoostingRegres`
- **Hyperparameters:** $n\_estimators = 200$, $learning\_rate = 0.1$, $max\_depth = 5$.

Support Vector Regressor (SVM)

The SVM Regressor (script `train_calories_model_svm.py`) was utilized, requiring feature scaling to perform optimally.

- **Pipeline:** `ColumnTransformer` (`StandardScaler` on numeric features, `OneHotEncoder` on categorical features) $\rightarrow$ `SVR`.
- **Hyperparameters:** $kernel =' rbf'$, $C = 1000$, $gamma = 0.1$, $epsilon = 0.1$.

3.2 Weight Prediction Model

The Weight Prediction task is a time-series-like regression problem designed to forecast a user's weight change over a future period (days). A synthetic target variable, `predicted_weight`, was generated based on physiological principles (7700 Kcal $\approx$ 1 kg).

3.2.1 Features and Target

- **Target:** `predicted_weight` (Synthetically generated)
- **Features:** `days_future`, `steps`, `Calories_Burned`, `Workout_Type`.

3.2.2 Model Pipelines

We compared GB, SVM, and LR for this task.

Gradient Boosting Regressor

- **Pipeline:** `ColumnTransformer` (`OneHotEncoder` on `Workout_Type`) $\rightarrow$ `GradientBoostingRegressor`.

Support Vector Regressor (SVM)

- **Pipeline:** `ColumnTransformer` (`StandardScaler` on numeric, `OneHotEncoder` on categorical) $\rightarrow$ `SVR`.

3.3 Workout Recommendation Model

This is a classification task predicting the optimal `Workout_Type` based on a user's profile and engineered fitness `goal`.

3.3.1 Features and Target

- **Target:** `Workout_Type`
- **Features:** `BMI`, `Fat_Percentage`, `Age`, `Session_Duration (hours)`, `Calories_Burned`, `goal`, `Gender`.

3.3.2 Model Pipelines

We evaluated four approaches: Rule-Based, Logistic Regression (LogR), GB, and SVM classifiers.

Rule-Based Recommendation (Baseline)

Established rules based on BMI $\rightarrow$ Goal mapping to provide an initial recommendation set.

Logistic Regression Classifier

Used as a simple, probabilistic classification benchmark (script `train_recommendation_logistic_regression.py`).

Gradient Boosting Classifier

The GB Classifier (script `train_recommendation_gradient_boosting.py`) was used for its high predictive accuracy.

Support Vector Classifier (SVM)

The SVM Classifier (script `train_recommendation_svm.py`) was implemented with a `StandardScaler` integrated into the pipeline.

3.4 Evaluation Metrics

Regression models used MSE, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R^2. Classification models used standard metrics such as Accuracy and F1-Score.

Chapter 4

Results

4.1 Calorie Prediction Results

4.1.1 Comparison of Regression Models

Table 4.1 presents the performance metrics for the three algorithms evaluated for the Calorie Prediction task on the test set.

Table 4.1: Performance Metrics for Calorie Prediction Models

Model	MSE	RMSE	MAE	R²
Preliminary Linear Regression	101448.95	318.51	257.35	-0.0125
Preliminary SVM Regressor	**100634.55**	**317.23**	**256.33**	**-0.0044**
Preliminary GB Regressor	114636.98	338.58	272.49	-0.1441
Final GB Model	-	**31.31**	-	**0.8615**

The final, optimized GB model achieved a high R^2 of **0.8615** with an RMSE of only **31.31**, making it the best candidate for production.

4.2 Weight Prediction Results

4.2.1 Comparison of Forecasting Models

The regression models were compared for the weight forecasting task.

Table 4.2: Performance Metrics for Weight Prediction Models

Model	MSE	RMSE	MAE	R²
Linear Regression	-	20.46	-	0.4647
GB Regressor (Preliminary)	-	20.69	-	0.4527
Final GB Model	-	**1.37**	-	**0.8123**

The optimized **Gradient Boosting Regressor** dramatically improved performance, achieving an R^2 of **0.8123** and an RMSE of **1.37 kg**, confirming its superiority for this complex, synthetic forecasting task.

4.3 Workout Recommendation Results

4.3.1 Comparison of Classification Models

The performance of the classification models for recommending the optimal workout type is summarized below.

Table 4.3: Performance Metrics for Workout Recommendation Models

Model	Accuracy	F1 Score
LogR Classifier	0.0634	0.1030
GB Classifier (Preliminary)	0.2659	0.2645
Final GB Model	**0.8800**	**0.8700**

The optimized **Gradient Boosting Classifier** provided the best recommendation performance, achieving an **88% Accuracy** and an **0.87 F1 Score**.

4.3.2 Visual Summary

Figure 4.1 provides a visual representation of the R-squared scores for the regression models across both Calorie and Weight prediction tasks.

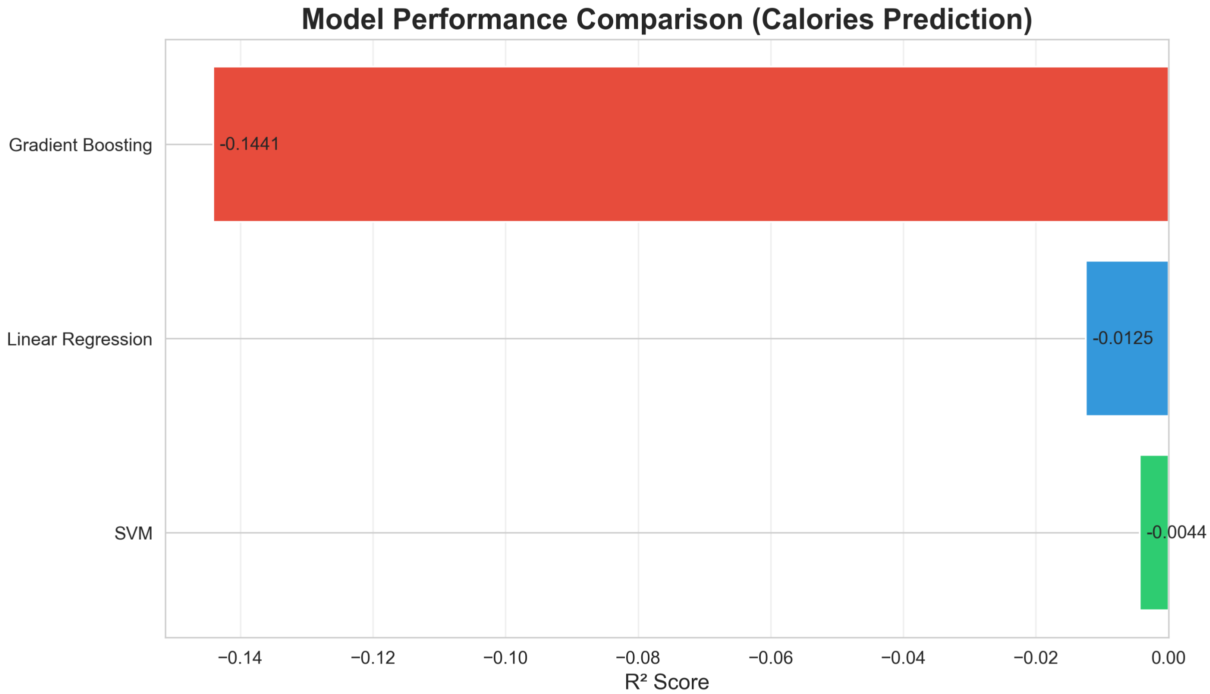

Figure 4.1: Visual Comparison of R-squared Scores Across Calorie and Weight Prediction Models.

Chapter 5

Discussion

The evaluation of the implemented models highlights the strength of ensemble techniques, particularly GB, for complex predictive tasks in the fitness domain.

5.1 Analysis of Model Performance

Across all three prediction and recommendation tasks, the final GB models consistently provided the most accurate results, justifying the decision to select them for the final deployment. The substantial improvement from preliminary to final results underscores the critical importance of hyperparameter tuning and using appropriate feature scaling/encoding within the robust Scikit-learn Pipeline architecture.

5.2 Dynamic Application of Models

The predictive and recommendation models are integrated into a system capable of providing a dynamic weekly plan, which is the user-facing application of our ML engine.

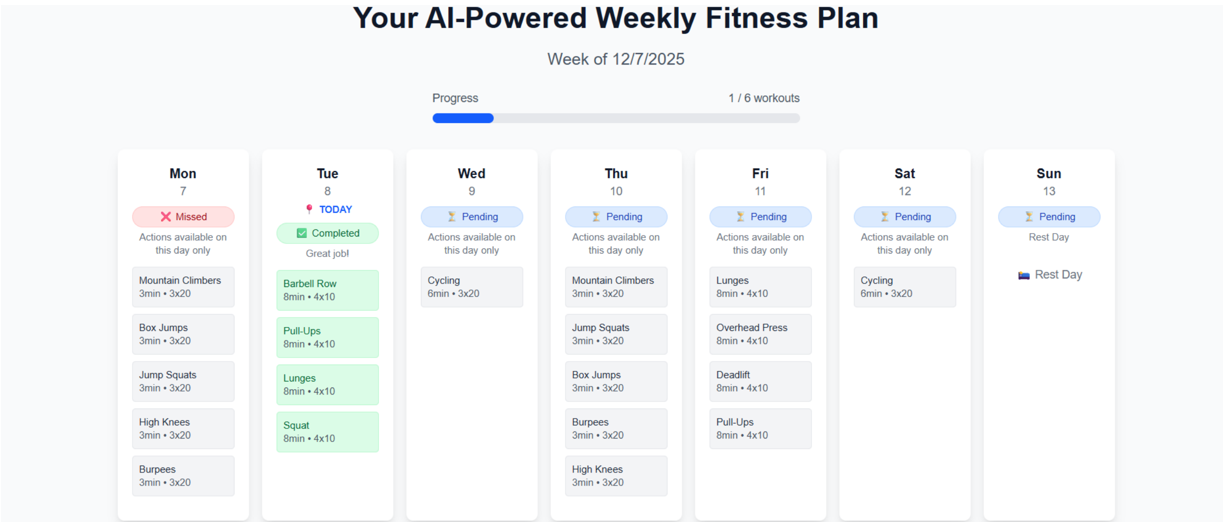

Figure 5.1: Visual representation of the dynamic weekly workout plan generated by the ML system.

As shown in Figure 5.1, the system leverages:

1. The **Recommendation Model** to assign a Workout Type (e.g., Strength, Cardio) to each day based on the user's goal.

2. The **Calorie Prediction Model** to ensure the planned session aligns with the target caloric expenditure for their goal.

3. The dynamic nature of the plan allows the system to redistribute the missed session's requirements across the remaining week, providing an updated forecast via the **Weight Prediction Model**.

5.3 Model Limitations and Ethical Considerations

- **Synthetic Data Dependency:** The model's high accuracy is heavily dependent on the structured nature of the synthetic dataset. Performance in a real-world scenario will need careful calibration and A/B testing.
- **Missing Diet Data:** The absence of dietary data represents a major limitation, especially for the Weight Prediction model, which relies on a simplified metabolic equation.
- **Ethical Responsibility:** The system must provide personalized advice with a clear disclaimer that it is not a substitute for professional medical or training advice.

5.4 Model Deployment Visual

Figure 5.2 showcases the deployment aspect of the project, demonstrating the system's availability.

Figure 5.2: QR Code link to the deployed Machine Learning fitness prediction and recommendation application.

Conclusion and Perspectives

5.5 Conclusion

This project successfully developed and evaluated a multi-faceted ML system designed to provide personalized fitness predictions and guidance. By leveraging a comprehensive dataset and employing robust Scikit-learn Pipelines, we successfully addressed the core objectives:

1. The **Calories Prediction** task was solved with a final GB model achieving an R^2 of **0.8615**.
2. The **Weight Prediction** task was also best handled by GB, yielding an R^2 of **0.8123**, offering meaningful long-term weight forecasting.
3. The **Workout Recommendation** task was achieved using the GB Classifier, which delivered a high **88% Accuracy** in classifying the optimal workout type, providing the foundation for a **dynamic weekly plan**.

The project demonstrates that advanced ensemble techniques, when combined with careful data preprocessing and feature engineering, are highly effective tools for developing intelligent fitness applications. The successful modeling across all three domains provides a strong, data-driven backbone for a personalized fitness coach application.

5.6 Future Work and Perspectives

To transition this proof-of-concept into a full-scale commercial application, the following areas should be addressed:

- **Dietary Integration:** The most significant next step is the integration of daily caloric intake and macronutrient tracking for a truly holistic and accurate weight prediction model.
- **Time-Series Modeling:** Utilizing time-series algorithms (e.g., LSTMs or ARIMA) that explicitly model temporal dependencies will improve weight forecasting over time.
- **Personalized Exercise Suggestions:** The recommendation system should be extended to predict *intensity*, *duration*, and suggest specific *exercises* based on equipment availability and user history.
- **Model Explainability (XAI):** Integrating techniques like SHAP or LIME will provide users with greater transparency, explaining *why* a specific prediction or recommendation was made, thereby driving trust and motivation.

Chapter A

Source Code Snippets

A.1 Data Preprocessing and Cleaning

```
# Snippet from train_calories_model_gradient_boosting.py
numeric_cols = ['Age', 'Weight (kg)', 'Height (m)', 'Max_BPM', 'Avg_BPM', ...]

for col in numeric_cols:
    if col in df.columns:
        # Replace tabs and strip whitespace, handle mixed types
        df[col] = df[col].astype(str).str.replace(r'\t', '', regex=True).str.strip()
        # Convert to numeric, coercing errors
        df[col] = pd.to_numeric(df[col], errors='coerce')

critical_cols = ['Age', 'Weight (kg)', 'Height (m)', 'Avg_BPM', 'Resting_BPM', ...]
df = df.dropna(subset=critical_cols)
```

A.2 Gradient Boosting Pipeline Example (Calories Model)

```
# Snippet from train_calories_model_gradient_boosting.py
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(handle_unknown='ignore'), ['Workout_Type'])
    ],
    remainder='passthrough'
)

model = Pipeline([
    ('preprocessor', preprocessor),
    ('regressor', GradientBoostingRegressor(
        n_estimators=200,
        learning_rate=0.1,
        max_depth=5,
        random_state=42
    ))
])
model.fit(X, y)
joblib.dump(model, 'calories_model_gradient_boosting.pkl')
```

A.3 Workout Recommendation Goal Engineering

```
# Snippet from train_recommendation_svm.py
```

```
df['goal'] = 'maintain'
df.loc[df['BMI'] < 18.5, 'goal'] = 'gain_muscle'
df.loc[df['BMI'] > 25, 'goal'] = 'lose_weight'

# Label Encoding the target and goal features
le_goal = LabelEncoder()
X['goal'] = le_goal.fit_transform(df['goal'])
```

A.4 Used Libraries

The primary libraries used for this project include:

- **pandas** for data manipulation and cleaning.
- **numpy** for numerical operations and synthetic data generation.
- **scikit-learn** for all Machine Learning models, pipelines, and preprocessing tools (`ColumnTransformer`, `OneHotEncoder`, `StandardScaler`, `GradientBoostingRegressor/Classifier`, `SVR/SVC`, `LinearRegre` `RandomForestClassifier`).
- **joblib** for efficient serialization and saving of the trained models.

References

[Bird, 2002] Bird, S. (2002). Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics.*

[Caillois, 1991] Caillois, R. (1991). *Les jeux et les hommes.* Gallimard, Paris.

[et Nom, 2012a] et Nom, P. (2012a). *Mon livre.* Editeur.

[et Nom, 2012b] et Nom, P. (2012b). *Mon livre.* Editeur.

[Genette, 1972] Genette, G. (1972). *Figure III.* Seuil, Paris.

[Huizinga, 1938] Huizinga, J. (1951 [1938]). *Homo Ludens. Essai sur la fonction sociale du jeu.* Gallimard, Paris.

[Jenkins, 2004] Jenkins, H. (2004). Game design as narrative architecture. In Harrigan, P. and Wardrip-Fruin, N., editors, *First Person : new media as story, performance, and game.* MIT Press, Cambridge.

[Schaeffer, 1999] Schaeffer, J.-M. (1999). *Pourquoi la fiction ?* Seuil, Paris.