University of Graz

The Micro-Degree "Artificial Intelligence and Society"

Predicting Marathon Finisher Time

Technical Aspect

Authors:
Emina Hamamdžić
Ajdin Kanlić
Maurice Mewes
Maximilian Schödlbauer

Reviewer:
Mgr. Štěpán Zapadlo
SFB PhD Student of the research group
Applied Mathematics and Machine Learning

June 5, 2026

Abstract

This project developed a machine learning pipeline to predict marathon finishing times using accessible training and physiological data. Working with a synthetic dataset of 100,000 runner profiles, we reduced the feature space to 23 variables and engineered key metrics, including `relative_experience` and `heart_efficiency` (a proxy for cardiac economy). Lasso regression identified baseline speed, experience, and cardiac efficiency as the primary independent predictors. Among six evaluated models (ranging from linear forms to neural networks), XGBoost achieved optimal performance with an MAE of 9.87 minutes and an R^2 of 86.23%. SHAP analysis validated that predictions align with sports science principles, highlighting personal best, cardiac economy, and injury history as dominant factors. Finally, an interactive `ipywidgets` dashboard was implemented for real-time validation.

1 Introduction and Problem Definition

The objective of this project was to develop a robust machine learning model to predict Marathon Finisher Time (MFT). Predicting performance over a 42.195 km race is a complex, non-linear problem driven by a combination of physiological parameters, training history, psychology, and external conditions. While professional sports analytics often relies on expensive laboratory testing (e.g., VO_2 max and blood lactate thresholds), our hypothesis is that MFT can be accurately predicted using easily accessible indicators and historical data from running clubs, commonly captured by everyday wearable tracking devices.

1.1 Project Settings and Goals

Having established the main goal of predicting MFT with easily accessible parameters, we searched for a fitting dataset for our project. For the analysis and model training, a massive dataset from kaggle was used. “Run Club Marathon Performance Dataset” is a synthetic dataset, created to simulate realistic marathon training performance data for machine learning practice. Inspiration for the dataset was “Can we predict marathon success from training data?”. The dataset was algorithmically generated using realistic correlations based on established sports science principles:

- Higher VO_2 max correlates with better performance
- Training consistency matters more than sporadic high-volume weeks
- Proper recovery, nutrition, and injury management impacts outcome
- Age, experience, and mental preparation play significant roles
- Weather and course difficulty affect race results

The purpose of the dataset design choices were:

- ML practitioners practicing multi-class classification with imbalanced data
- Data science students learning feature engineering and model evaluation
- Sports analytics enthusiasts exploring performance prediction models

- Beginners seeking a well-documented, realistic dataset

The dataset spanned a total of 40 features across 100,000 observations (marathon runners).

Table 1: Categorization and Overview of Marathon Runner Variables

Category	Variables & Descriptions
Demographics & Experience	`runner_id` (ID), `age`, `gender`, `bmi`, `running_experience_months`, `previous_marathon_count`, `personal_best_minutes`
Training Volume	`weekly_mileage_km`/`miles`, `runs_per_week`, `long_run_distance_km`, `speed_work_sessions_per_week`, `rest_days_per_week`
Adherence & Consistency	`training_program`, `training_adherence_pct`, `missed_workout_pct`, `consecutive_weeks_no_miss`, `training_streak_days`
Physiology & Health	`resting_heart_rate_bpm`, `vo2_max`, `recovery_score`, `injury_count`, `injury_severity`
Lifestyle & Habits	`sleep_hours_avg`, `nutrition_score`, `hydration_consistency`, `cross_training_hours_per_week`, `warmup_adherence_pct`, `stretching_adherence_pct`
Psychology & Environment	`motivation_level`, `mental_preparation_score`, `early_morning_run_frequency`, `weather_condition_training_pct`, `run_club_attendance_rate`
Race Day & Performance	`marathon_date`, `marathon_weather`, `course_difficulty`, `target_finish_time_minutes`, `actual_finish_time_minutes`, `goal_completion_rate`

2 Target Variable

Our target variable (Y) is a continuous value:

$$y = \texttt{actual\_finish\_time\_minutes}$$

It describes the time in minutes the observed participants needed to finish the marathon.

2.1 Mathematical Metrics and Equations

To evaluate the performance of all models and select the optimal, we tracked complementary statistical metrics:

2.1.1 Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

MAE provides a direct, linearly interpretable error expressed in minutes. An MAE of x describes that our model misses the actual result by x minutes on average. Unlike the Root Mean Squared Error, MAE is less sensitive to extreme outliers, providing a realistic representation of the model's average behavior.

2.1.2 Coefficient of Determination (R^2 Score)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

R^2 describes what percentage of the variance in the marathon finish times our model successfully explains using the input features. A value of, for instance, 0.8 indicates that our model captures 80% of the factors influencing the race, while the remaining 20% constitute unexplained noise.

2.1.3 Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Unlike MAE, RMSE squares individual errors before calculating the average, which means large deviations are penalized disproportionately. RMSE was therefore used as a stability detector: if the model predicted finishing times accurately for most runners but missed some by a large margin, MAE would remain relatively low whereas RMSE would spike drastically. It is thus desirable to achieve both low MAE and low RMSE simultaneously.

3 Data Analysis

Before any algorithm was applied, the data passed through a preprocessing pipeline to ensure the mathematical stability of the models. Feature engineering and algorithmic feature selection are described separately in Section 4, as they build upon the cleaned and encoded data produced in this section.

3.1 Data Cleaning

3.1.1 Rough Feature Selection — Expertise and Plausibility

Firstly, all features were thoroughly examined to determine which have a plausible and possibly measurable effect on the target variable. For this, sports scientific expertise and literature research were surveyed. This resulted in 22 predictor features remaining for further analysis, plus the target variable `actual_finish_time_minutes`, yielding 23 entries in total:

Table 2: Selected features after expertise and plausibility check.

Feature	**Description**
age	Age in years
gender	Participant gender
running_experience_-months	Running experience in months
previous_marathon_-count	Number of previously completed marathons
personal_best_minutes	Personal best marathon time in minutes
weekly_mileage_km	Weekly running mileage in kilometers
runs_per_week	Number of runs per week
long_run_distance_km	Longest weekly run distance in kilometers
speed_work_sessions_-per_week	Number of speed work sessions per week
cross_training_hours_-per_week	Weekly cross-training duration in hours
resting_heart_rate_bpm	Resting heart rate in beats per minute
vo2_max	Maximal oxygen uptake
bmi	Body mass index
sleep_hours_avg	Average sleep duration in hours
injury_count	Number of injuries
injury_severity	Injury severity score
missed_workout_pct	Percentage of missed workouts
goal_completion_rate	Goal completion rate
warmup_adherence_pct	Warm-up adherence percentage
stretching_adherence_pct	Stretching adherence percentage
marathon_weather	Weather conditions during the marathon
course_difficulty	Marathon course difficulty rating
actual_finish_time_min-utes	Actual marathon finish time in minutes (target vari-able)

To gain a better understanding of the distribution of these remaining features, histograms were plotted for each feature.

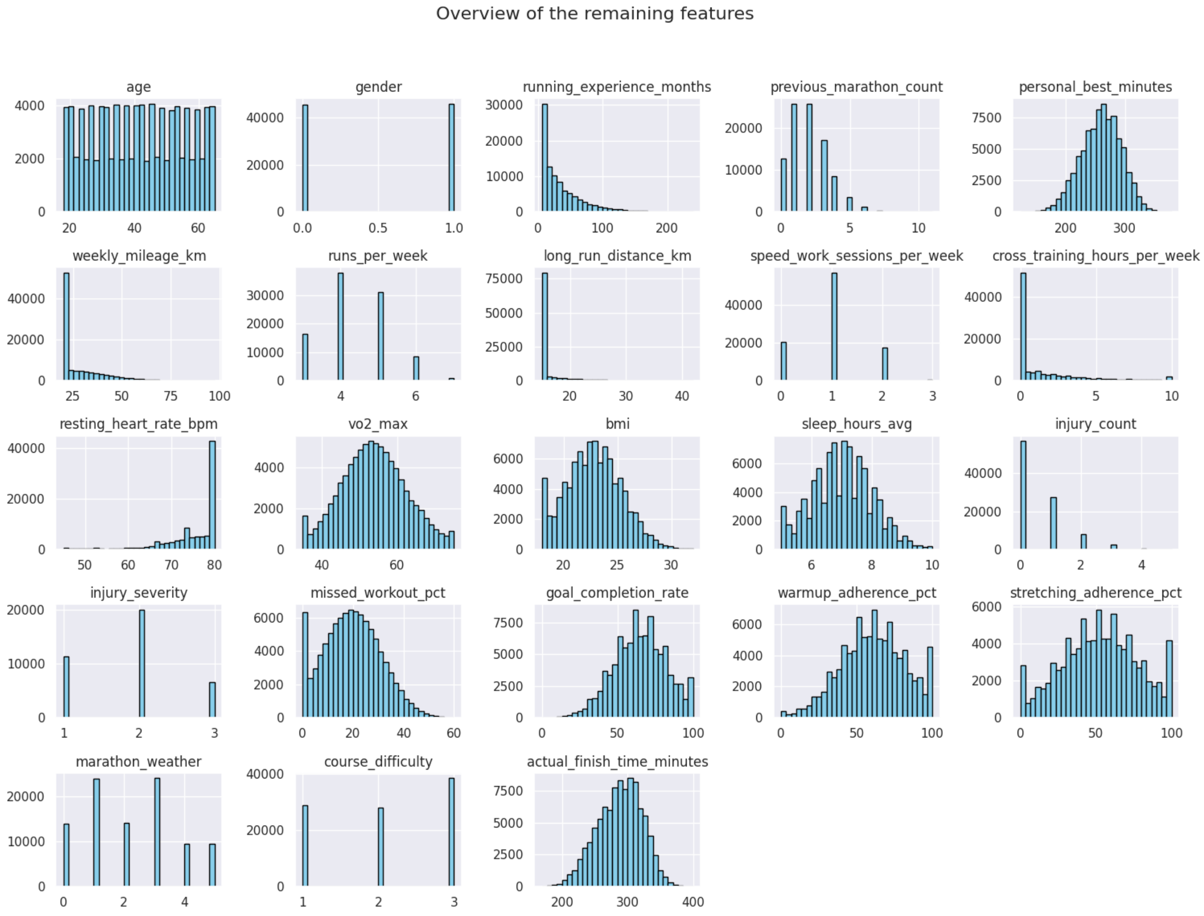

Figure 1: Histogram of remaining features.

3.2 Handling Missing Values

To determine the quality of the data, missing values were calculated and accounted for.

3.2.1 actual_finish_time_minutes (2,500 missing)

Descriptive statistical analysis revealed that `actual_finish_time_minutes` had 97,500 populated rows out of a total of 100,000. As 2.5% of the data suffered from missing entries, and filling these through imputation would distort the true distribution of values, the 2,500 affected rows were excluded from further examination.

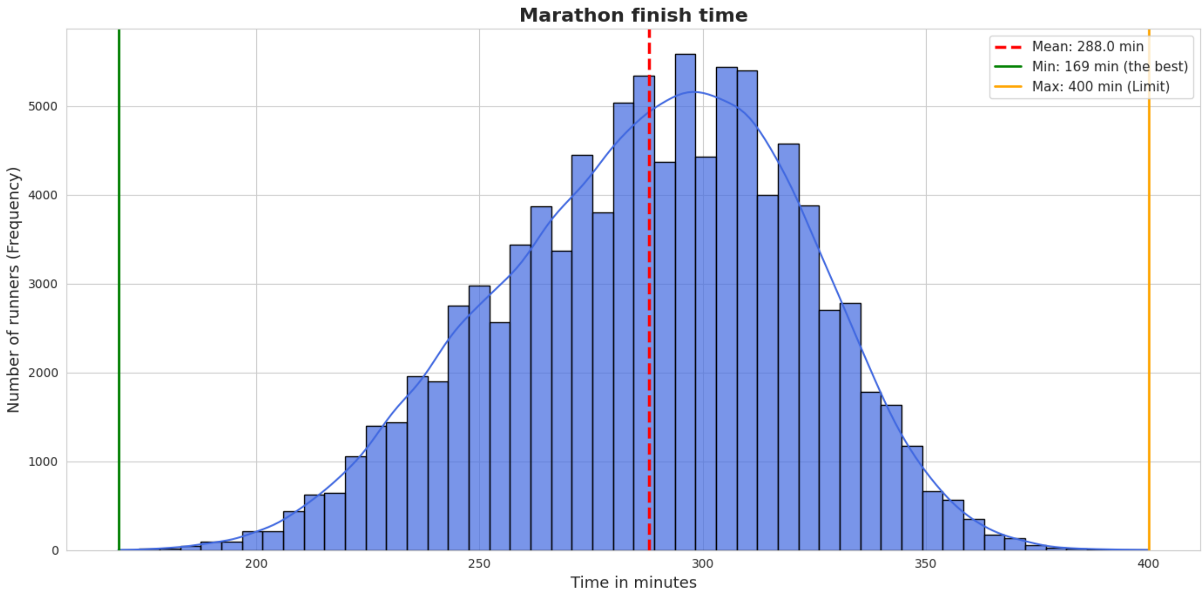

Figure 2: Histogram of marathon finisher times.

3.2.2 cross_training_hours_per_week (10,000 missing)

As can be seen in Figure 1, the distribution of `cross_training_hours_per_week` within the running community was found to be highly right-skewed. The vast majority recorded very few hours while there were some extreme values. Imputation by mean or median is not plausible with this distribution scheme, because it would destroy the natural variance. Random sampling was therefore utilized: by randomly drawing 10,000 values from the already existing data, the natural right-skew was preserved and the missing values accordingly imputed.

3.2.3 personal_best_minutes (13,416 missing)

Having a personal best time before a marathon is highly dependent on motivation and training. Recreational runners often have better or worse personal bests compared to race results. The Gaussian distribution was preserved by also using random sampling to impute this feature.

3.2.4 sleep_hours_avg (5,000 missing)

Human sleep cycles naturally appear in discrete intervals (6, 7, 8, or 9 hours). Using mean or median as an imputation method would not preserve the natural distribution of the feature. Again, random sampling was used to maintain the Gaussian curve of sleep patterns.

3.2.5 injury_severity (59,190 missing)

Instead of dropping this column or imputing missing data arbitrarily, domain knowledge from sports science was applied. It is most likely that runners leaving this column empty are healthy and do not suffer from any injury. Therefore, the missing values were imputed with 0.

3.2.6 vo2_max (12,000 missing)

Filling in the missing values for `vo2_max` was not as straightforward as for the other features. `vo2_max` describes a vital physiological indicator and is a stable parameter when analyzing physical fitness and performance capability. Rather than relying on random sampling, rudimentary machine learning was used for imputation.

Firstly, correlation with other features was analyzed using Spearman correlation, targeting `vo2_max` against all other numeric attributes. A very strong negative correlation ($r = -0.877$) was found with `resting_heart_rate_bpm`. Low correlations ($r > 0.1$, range $[-0.21, 0.19]$) were found with `running_experience_months`, `personal_best_minutes`, `age`, and `actual_finish_time_minutes`. All other tested features showed a correlation below $r = 0.1$ and are thus not of interest for further estimation. After visually verifying these linear structures with scatter plots, predictive linear modeling was deemed a viable option. For predictor selection, two approaches were used: AIC Stepwise Selection and LassoCV.

Approach A: The mathematical quality criterion for stepwise selection is the Akaike Information Criterion (AIC):

$$AIC = 2k - 2\ln(\hat{L})$$

Because no automated stepwise algorithm is available in Python, a custom one was implemented. The stepwise algorithm evaluates every potential feature, dynamically executing a forward step (adding a predictor) or a backward step (removing a predictor). The AIC is calculated after every iteration; when no further improvement occurs, the feature selection is concluded.

Approach B: LassoCV (Least Absolute Shrinkage and Selection Operator with 5-fold cross-validation) adds an L1 penalty to the ordinary least squares objective function:

$$\min_{\beta} \left(\frac{1}{2n} \|y - X\beta\|_2^2 + \alpha \|\beta\|_1 \right)$$

Lasso forces the coefficients of completely redundant or weak features to shrink exactly to zero, thereby performing variable selection and reducing model variance. The results can be seen in Figure 3.

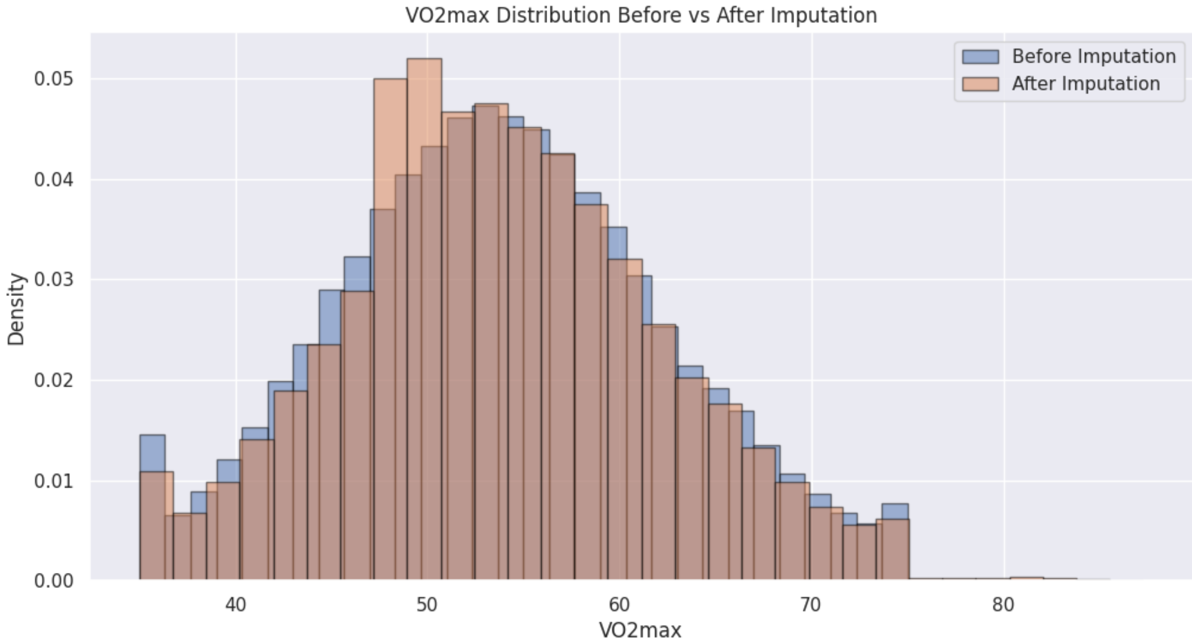

Figure 3: Histogram of `vo2_max` before and after imputation.

3.3 Encoding Categorical Variables

The dataset contains important categorical attributes such as `gender`, `training_program`, `marathon_weather`, and `course_difficulty`. Computational models exclusively operate on numerical values. The one-hot encoding technique was therefore used to transform categorical variables into binary columns, avoiding ordinal ranking — for instance, assigning 1, 2, 3 to course difficulty levels would imply that level 3 is three times more difficult than level 1, which is not the case.

3.4 Feature Scaling and Standardization

Numerical features such as `age`, `weekly_mileage_km`, `resting_heart_rate_bpm`, or `vo2_max` possess drastically different scales. `StandardScaler` was utilized to transform the data to have a mean $\mu = 0$ and a standard deviation $\sigma = 1$. This step is essential for linear models and neural networks, as these rely on Euclidean distance calculations and therefore require normalized feature scales.

4 Feature Engineering and Selection

4.1 Feature Engineering

Two new variables were created with the goal of converting raw numerical inputs into meaningful biological indicators.

4.1.1 relative_experience (Experience / Age)

To account for training experience in the context of biological age, the variable `relative_experience` was constructed. It places running tenure into direct relation with the runner's age. Two runners with identical experience in months will not perform equally if one is twenty years older than the other.

4.1.2 heart_efficiency (VO_2max / Resting Heart Rate)

While `vo2_max` indicates an individual's maximum oxygen capacity (the size of the engine), it does not reflect the cardiac workload required to achieve it. By dividing VO_2max by `resting_heart_rate_bpm`, `heart_efficiency` captures cardiac economy — a more informative indicator of endurance performance than either variable alone.

4.2 Feature Selection

Lasso regression was applied to identify and discard variables with negligible independent predictive contribution. Variables such as `gender` and `sleep_hours_avg` were penalized to zero, indicating that their variance is already implicitly captured by the retained predictors — biological sex differences in aerobic capacity, for instance, are largely reflected in `heart_efficiency` and `personal_best_minutes`. This regularization step reduced model complexity and mitigated the risk of overfitting, improving expected generalization to unseen data.

The resulting three-variable model adheres to the principle of parsimony: each retained feature represents a distinct and non-redundant performance dimension:

- Physiological engine: `heart_efficiency`
- Experience base: `running_experience_months`
- Baseline speed capacity: `personal_best_minutes`

5 Model Development

The predictive models were developed using a paradigm of incremental complexity, traversing linear forms, parametric networks, and non-linear ensembles to optimize performance while avoiding overfitting. To establish a reference benchmark under identical preprocessing conditions, the baseline Linear Regression model was fitted on the full 20-dimensional feature space prior to Lasso-based feature selection. The architecture, evolution, and diagnostics of each applied model are detailed below.

5.1 Linear Regression (Baseline)

An Ordinary Least Squares (OLS) model served as the baseline, fitting a hyperplane across the 20-dimensional feature space by minimizing the Residual Sum of Squares (RSS).
Performance: MAE $\approx$ 12.00 min | RMSE $\approx$ 15.20 min | $R^2 \approx 80.19\%$
Learning curve analysis indicated high bias (underfitting): training and test error converged to a consistently elevated plateau. Given that key physiological phenomena in marathon running — such as glycogen depletion kinetics beyond kilometer 30 or the non-linear effect of heat and humidity on thermoregulation — exhibit inherently non-linear behavior, the strict linearity assumption of OLS proved insufficient to capture the underlying data structure.

5.2 Polynomial Regression (Degree = 2)

To introduce non-linearity while retaining coefficient interpretability, the feature space was expanded via `PolynomialFeatures(degree=2)`, generating quadratic terms (X_i^2) and pairwise interaction terms ($X_i \times X_j$).
Performance: MAE $\approx$ 10.29 min | RMSE $\approx$ 12.85 min | $R^2 \approx 85.12\%$
The substantial improvement over the baseline confirmed the presence of meaningful non-linear interactions (e.g., between age and injury history). Bias-variance balance remained acceptable; however, the quadratic expansion introduces a risk of dimensionality explosion and increased sensitivity to outliers.

5.3 Artificial Neural Network (MLPRegressor)

A fully connected feedforward network with three hidden layers (128 $\rightarrow$ 64 $\rightarrow$ 32 neurons), ReLU activation, and Adam optimization (learning rate = 0.001) was trained as a deep learning benchmark.
Performance: MAE $\approx$ 11.15 min | RMSE $\approx$ 14.10 min | $R^2 \approx 82.50\%$
The model exhibited high variance (overfitting): training error was substantially lower than test error. Dense architectures tend to memorize row-specific noise in structured tabular datasets where interpretable physiological correlations dominate, rather than learning generalizable patterns — making this architecture a suboptimal fit for the present data.

5.4 Support Vector Regressor (SVR)

An SVR with a Radial Basis Function (RBF) kernel was implemented, implicitly projecting data into a high-dimensional Hilbert space. The ε-insensitive loss function (ε-tube) was configured to prioritize prediction stability over minimizing marginal errors.
Performance: MAE $\approx$ 10.67 min | RMSE $\approx$ 13.40 min | $R^2 \approx 84.02\%$
Despite the RBF kernel's capacity to model complex non-linear boundaries, performance was sub-optimal. The likely cause is the presence of sparse binary columns introduced by one-hot encoding of categorical variables: Euclidean distance metrics, on which RBF kernels rely, lose discriminative power in high-dimensional sparse binary spaces.

5.5 Random Forest Regressor

A bagging ensemble of parallel decision trees was constructed, each trained on a bootstrap sample of the data. Feature subspacing (random feature subset per split) was applied to decorrelate individual trees, with final predictions obtained by averaging across the ensemble.
Performance: MAE $\approx$ 10.48 min | RMSE $\approx$ 13.10 min | $R^2 \approx 84.73\%$
The model demonstrated strong robustness to outliers and low variance. However, individual trees exhibited a tendency toward excessive depth, resulting in overfitting to narrow runner sub-profiles. This left systematic residual error that sequential learning methods are better positioned to address.

5.6 XGBoost (Extreme Gradient Boosting)

XGBoost was selected as the final model. Unlike Random Forest, which constructs trees independently in parallel, XGBoost builds trees sequentially, with each tree minimizing the residual gradient of its predecessor. Hyperparameter optimization was performed via

RandomizedSearchCV with 5-fold cross-validation, yielding the following configuration: a low learning rate of 0.05 for stable convergence, a maximum tree depth of 6 to limit model complexity, and stochastic subsampling of 80% at both the row (subsample) and feature level (colsample_bytree) as an additional regularization mechanism.
Performance: MAE = 9.87 min | RMSE = 12.15 min | R^2 = 86.23%
Stochastic subsampling at the row and feature level acted as an effective regularization mechanism, preventing the model from fitting dataset-specific anomalies and instead enforcing learning of globally stable performance patterns. Combined with XGBoost's native L1/L2 regularization, this yielded the best generalization across all tested architectures, explaining 86.23% of total variance in finishing times with a mean absolute error below 10 minutes.

5.7 Model Interpretation (SHAP Values)

Although XGBoost is inherently a black-box model, SHAP (SHapley Additive exPlanations) analysis was applied to examine the contribution of each individual feature to the model's predictions. The analysis revealed three key findings:
Highest impact: As expected, personal_best_minutes possesses the highest predictive power, reflecting baseline speed capacity and accumulated race experience.
Physiological pillar: The cardiac efficiency index heart_efficiency directly drives predictions toward faster finishing times when its value is optimal, confirming its role as the primary physiological predictor in the model.
Risk factors: The number and severity of historical injuries (injury_severity) emerge as strong non-linear factors that substantially prolong race completion time.
These findings confirm that the model has not learned spurious correlations, but has reconstructed established laws of sports physiology and endurance performance within its mathematical weights.

5.8 Model Comparison

The following table summarizes the results of all the explained Models.

Table 3: Model comparison

Model	**MAE (min)**	**RMSE (min)**	$\mathbf{R^2}$	**Key diagnostic**
Linear regression	12.00	15.20	80.19%	High bias – underfitting; cannot capture non-linear physiological relationships
Polynomial regression (deg. 2)	10.29	12.85	85.12%	Good fit; risk of dimensionality explosion and outlier sensitivity
Neural network (MLP)	11.15	14.10	82.50%	High variance – overfitting; dense layers memorize noise on tabular data
SVR (RBF kernel)	10.67	13.40	84.02%	RBF distance metric loses discriminative power with sparse binary features
Random forest	10.48	13.10	84.73%	Robust to outliers; individual trees overfit narrow runner sub-profiles
XGBoost	**9.87**	**12.15**	**86.23%**	Sequential boosting + stochastic subsampling + L1/L2 regularization; best generalization

6 User Interface for Runners

To make the model's predictive capabilities accessible, a live prediction dashboard was implemented within the `.ipynb` notebook using the `ipywidgets` library. The interface features interactive sliders that automatically load real marathoner profiles from the test dataset. By activating the integrated button, the user can observe a real-time comparison between a runner's actual finishing time and the prediction generated by the tuned XGBoost pipeline, allowing on-the-spot accuracy evaluation.

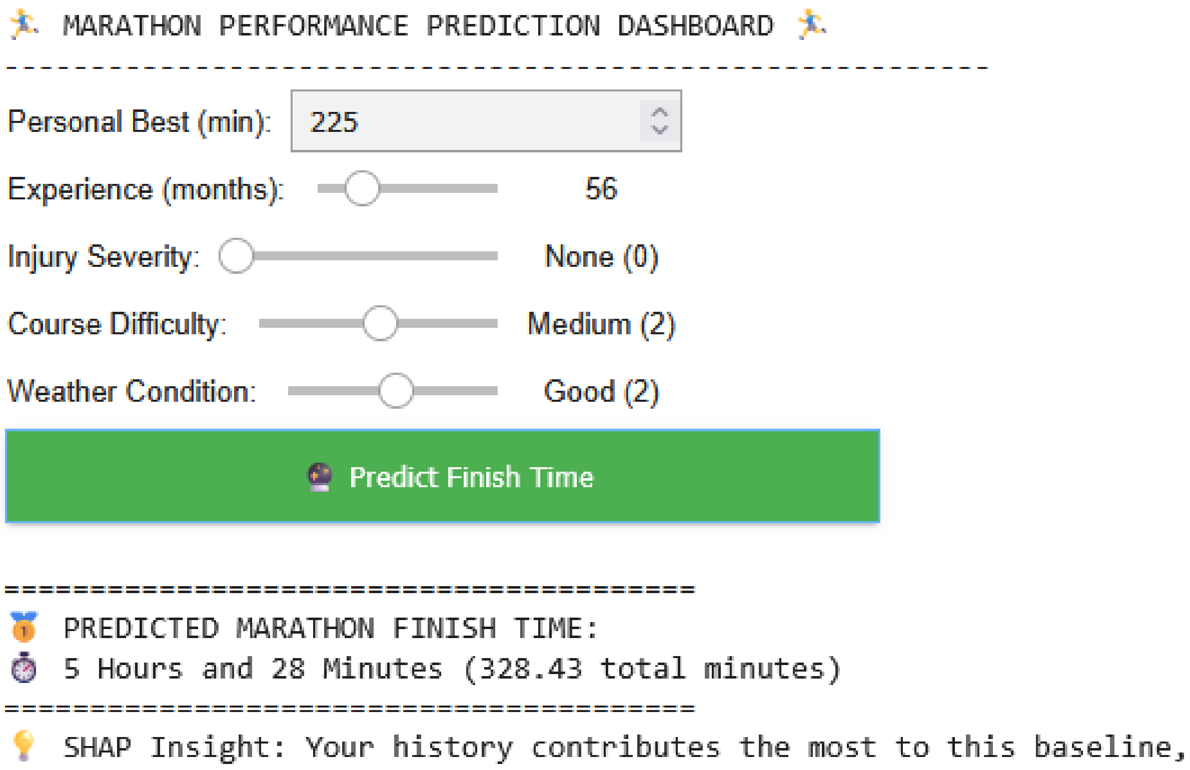

Figure 4: User Interface of the predictor.

7 Conclusion

This project successfully designed and implemented an end-to-end machine learning workflow to predict Marathon Finisher Time (MFT) using non-invasive, easily obtainable wearable metrics. Through robust feature engineering and Lasso-driven selection, we successfully isolated the primary variables dictating long-distance athletic performance, demonstrating that a condensed multi-dimensional profile—anchored by physiological efficiency (`heart_efficiency`), historical racing speed, and training consistency—is highly predictive of absolute race outcomes.

Among the six evaluated architectures, the XGBoost model proved superior, accounting for 86.23% of the overall variance with an exceptionally low mean absolute error of 9.87 minutes. This performance highlights the capability of gradient-boosting ensembles to effectively map highly non-linear constraints, such as cumulative physiological fatigue and metabolic thresholds, which traditional linear regression architectures fail to represent. Crucially, the subsequent SHAP analysis mathematically validated that our black-box model accurately prioritized parameters that strictly align with established principles of exercise physiology. Finally, the creation of an interactive dashboard demonstrates how complex statistical predictors can be translated into practical toolkits for recreational athletes and coaches to adjust training volume, manage recovery, and set optimal, data-driven pacing strategies prior to race day.

References

[1] Coenen, P., et al. (2024). Machine learning-based estimation of VO2 max using wearable sensor data in real-world running scenarios. *npj Digital Medicine*, 7(1), 42–51.

[2] Loreto, V., Marini, L., & Petrolo, G. (2022). Predicting marathon performance using wearable device data and machine learning. *Journal of Sports Analytics*, 8(2), 115–127.

[3] Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32.

[4] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).