Decoding Endurance Data: Predictive Modelling of Performance and Workout Classification

Anouk Schnidrig, Jonas Senn, Anastasiia Obidonova

December 3, 2025

Abstract

This study analyzes sports data collected over five consecutive seasons to evaluate the predictive accuracy of a performance model and to assess how effectively classification algorithms can differentiate between distinct types of workouts. The dataset was manipulated and analyzed in Python to uncover correlations between physiological performance and training metrics. The aim of this work is to demonstrate how personal sports tracking data can be used to derive meaningful insights into individual performance and training behavior across different conditions.

1 Research Questions

Based on the recorded data spanning five seasons, the following research questions were formulated:

How accurately can a regression model predict the moving time of a run using elapsed distance as its primary feature? The aim of this question is to assess how increasing distance influences the average running time as an indicator of endurance performance.

How effectively can a classification model differentiate between running, swimming, and cycling sessions when using distance and maximum speed as the primary predictive features? This question explores how well simple training features support automatic recognition of running, swimming, and cycling sessions. Previous research has demonstrated that various machine-learning approaches can accurately differentiate athletic movements using sensor-derived or performance-based features (Obradović & Kečo, 2024). In this case study, we decided to train a supervised multi-class classification model. Such an approach can be refined for enhancing training analysis.

2 Methodology

2.1 Data Collection

The dataset used in this study was collected using a GARMIN Forerunner 965 sports watch, which continuously recorded sports activities and physiological metrics such as heart rate, distance, pace, caloric output (energy expenditure), elevation as well as other metrics. Data collection took place between December 7, 2020 and October 29, 2025, covering training sessions across five consecutive seasons. The recorded data was synchronized and shared onto Strava platform[1], from which a CSV export[2] of the athlete's activities was obtained for further processing and analysis.

2.2 Data Wrangling

Reprocessing Data for Linear Regression Model After exporting the activity data from Strava as a CSV file, the dataset underwent filtering and formatting directly in Python. The preprocessing pipeline began by importing a restricted subset of variables using the usecols argument in pd.read_csv(), wrapped in a try/except FileNotFoundError block to handle read errors. Distance values recorded with commas (e.g., "1,600") were standardized to a period-separated format via df["Distance"].str.replace(",", "."), after which the field was cast to numeric form using astype(float). Moving time was normalized by converting seconds to minutes (df["Moving Time"] = df["Moving Time"] / 60) to ensure consistent units across analyses. A missing-value check (isnull().sum()) verified dataset completeness prior to modeling. Finally, the dataset was filtered to retain only running sessions (df[df["Activity Type"] == "Run"]), ensuring that the regression model was trained on a homogeneous activity subset. This combination of controlled column selection, error-handled data loading, string normalization, type conversion, and targeted filtering produced a clean, structured dataset suitable for statistical modeling.

Reprocessing Data for Classification Model The preprocessing workflow began by importing only the relevant variables using the usecols parameter in pd.read_csv(), embedded in a try/except FileNotFoundError block to ensure robust file handling. As in the regression pipeline, distance entries containing commas were standardized using df[”Distance”].str.replace(”,”, ”.”) and subsequently cast to floating-point values through astype(float). Moving time values were normalized by converting seconds to minutes (df[”Moving Time”] = df[”Moving Time”] / 60) to maintain consistent temporal units across activities. Activity-specific filtering was then applied: separate dataframes for running, swimming, and cycling were generated using df[df[”Activity Type”].isin([...])]. Additional cleaning steps included correcting implausible swimming distances (lambda x: x/1000 if x ¿ 50 else x) and removing cycling observations with erroneous zero-distance entries. The cleaned subsets were concatenated into a unified dataframe (pd.concat()), followed by a missing-value assessment using isnull().sum(). Finally, the categorical activity labels were transformed into numeric format using LabelEncoder(), preparing the dataset for downstream model training. Together, these procedures ensured a structured, validated, and uniformly formatted dataset suitable for multi-class classification analysis.

2.3 Data Mining

Model Training for Linear Regression The cleaned running dataset was used to train a univariate linear regression model, with distance as the predictor and moving time as the response. The data was split into training and testing subsets using train_test_split() (80/20 ratio, fixed random state), after which the model was fitted using scikit-learn’s LinearRegression() implementation. Predictions for the test set were generated via model.predict(X_test). The model’s performance was evaluated using mean absolute error, mean squared error, root mean squared error and the coefficient of determination. Finally, the fitted regression line and the data points were visualized with matplotlib to illustrate model fit and predictive accuracy.

Model Training for Classification The cleaned dataset was split into training and testing subsets using train_test_split() with a 60/40 ratio and a fixed random state. Three classifiers—a RandomForestClassifier with 500 estimators, a DecisionTreeClassifier, and a KNeighborsClassifier with k = 5—were instantiated and trained iteratively via .fit(X_train.values, y_train.values) to learn patterns in distance and moving time across the three activity types. Their performance was then evaluated by predicting on the test set and computing macro-averaged F1-scores, providing a balanced assessment of precision and recall. For demonstration, a synthetic test activity was passed through each model, and the predicted labels were extracted with idxmax() and mapped back to their categorical form using the fitted LabelEncoder to obtain an interpretable activity classification.

3 Results

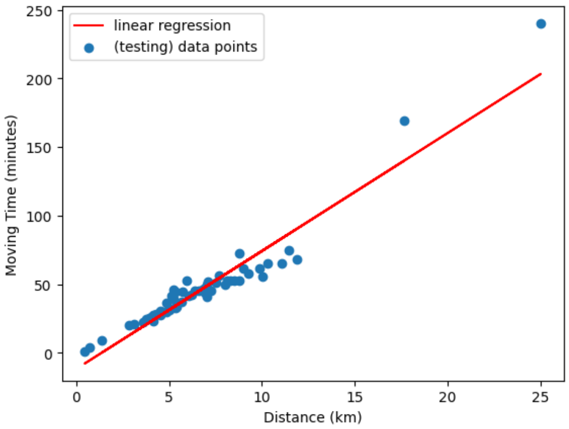

Figure 1: Linear Regression – Moving Time over Distance

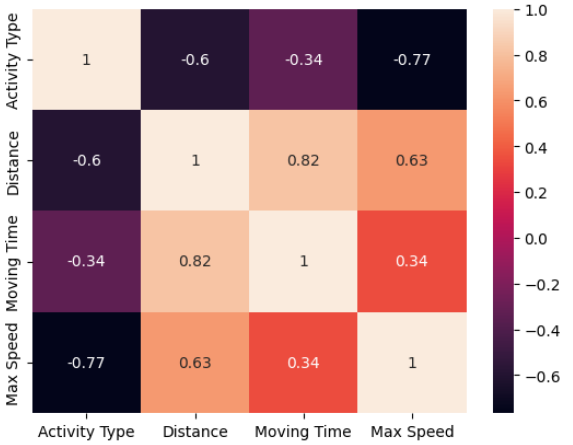

Figure 2: Correlation Heatmap

```
F1_score of Random Forest Classifier: 0.9337682201600291
F1_score of Decision Tree Classifier: 0.9360318228242757
F1_score of K-Neighbors: 0.9455156798610629
```

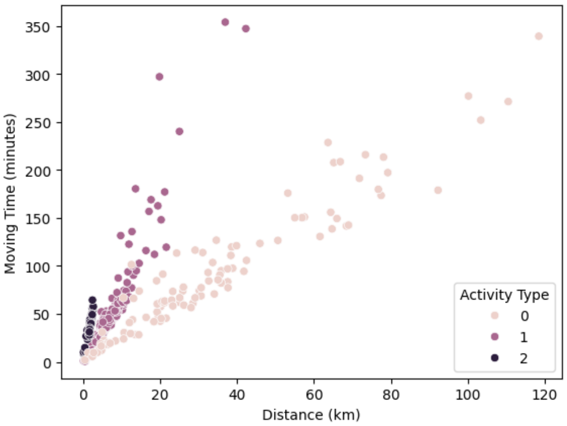

Figure 3: Activity Distribution by Distance and Moving Time

4 Discussion and Model Evaluation

Linear Regression: The regression model shows a strong linear relationship between distance and moving time. The coefficient (8.59) indicates that each additional kilometer increases predicted running time by an average of 8.6 minutes, which closely aligns with typical endurance running patterns, such as Peter Riegel's Race Time Prediction (Riegel, 1981)[4]. However, Riegel expects the relationship to be slightly curved (a power law with exponent 1.06), while our model assumes a straight line. Especially for longer distances (e.g., 18–25 km), the linear model underestimates the slowdown, which is exactly what Riegel's formula predicts. **MAE:** 6.41 minutes MAE reflects the average absolute prediction error. An error of about 6 minutes is small relative to the full range of running times, indicating good practical accuracy. **MSE:** 91.26 / **RMSE:** 9.55 minutes MSE penalizes larger errors more strongly, while RMSE expresses this penalty in minutes. An RMSE under 10 minutes suggests only moderate deviations and few large outliers. **R^2:** 0.92 The model explains about 92% of the variance in moving time, which is notably high for a single-feature model. A similarly high linear regression was found in a study of recreational marathon runners (Smyth & Muñiz-Pumares, 2020)[5]. This suggests that distance is the dominant factor determining run duration in this dataset. Overall Together, the metrics and plot show that the model predicts running time accurately for most distances, with only a few longer or atypical runs deviating more strongly. Despite its simplicity, the model performs very well and captures the essential relationship between distance and moving time.

Classification: The classification models achieved consistently high F1-scores, with values of approximately 0.94 for both the Random Forest and Decision Tree classifiers and 0.95 for the K-Neighbors classifier. These results indicate that all three models differentiate effectively between runs, swims, and bike rides based solely on distance and moving time. A related study by Fazli et al. (Fazli et al., 2020)[6], using a hierarchical neural-network–based activity recognition approach, reported similarly high classification performance with an accuracy of approximately 0.93. The visualizations support these outcomes: the scatterplot shows clear separation between activity types, and the heatmap illustrates strong correlations — particularly between distance and moving time — which help the models form accurate decision boundaries. Swimming sessions cluster at very short distances, running activities concentrate in a moderate range, and cycling covers the longest distances and durations. This structured distribution enables even relatively simple models to classify the activities with high reliability. Overall, the results demonstrate that the selected features provide sufficient discriminatory power for accurate multi-class classification, and the consistently high F1-scores confirm robust model performance across all three classifiers.

References

[1] Fabio Kaufmann Athlete Profile on Strava: https://strava.app.link/etLq8oUALYb

[2] GitHub Repository: Decoding Endurance Data: https://github.com/missfaypy/Decoding-endurance-Data/tree/main

[3] Obradović, A., & Kečo, D. (2024). Sports results prediction model using machine learning. SAR Journal - Science and Research, 184–189. https://doi.org/10.18421/sar73-03

[4] Riegel, P. S. (1981). Athletic records and human endurance. American Scientist, 69(3), 285–290.

[5] Smyth, B., & Muniz-Pumares, D. (2020). Calculation of Critical Speed from Raw Training Data in Recreational Marathon Runners. Medicine and science in sports and exercise, 52(12), 2639. https://doi.org/10.1249/MSS.0000000000002412

[6] Fazli, M., Kowsari, K., Gharavi, E., Barnes, L., & Doryab, A. (2020). HHAR-net: Hierarchical Human Activity Recognition using Neural Networks. arXiv. https://doi.org/10.48550/arXiv.2010.16052