Blue text on a black background

Description automatically generated

**College of Engineering, Environment, and Computing**

**School of Science**

**BSc Computer Science**

**6001CEM Computing Individual Research Project**

**Project Report**

**Predicting Competitive Swimming Performance Using Machine Learning Models**

By

**Muhammad Abdullah**

**SID: 13115676**

**1st Supervisor: Amanda Brooks**

**2nd Supervisor: <your second supervisor’s name>**

Submitted in partial fulfilment of the requirements for the Degree of Bachelor of Science in BSc Computer Science

Academic Year: 2024/25

**Declaration of Originality**

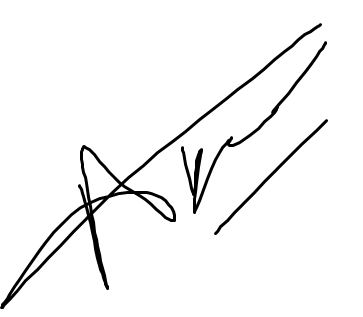
I declare that this project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

**Statement of copyright**

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to Coventry University. Support, including funding, is available to commercialize products and services developed by staff and students. Any revenue that is generated is split with the inventor/s of the product or service. For further information please see [www.coventry.ac.uk/ipr](http://www.coventry.ac.uk/ipr) or contact [ipr@coventry.ac.uk](mailto:ipr@coventry.ac.uk).

**Statement of ethical engagement**

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (https://ethics.coventry.ac.uk/) and that the application number is listed below (Note: Projects without an ethical application number will be rejected for marking).

Signed: 

Date: 10/04/2025

Please complete all fields.

|  |  |
| --- | --- |
| First Name: | **Muhammad** |
| Last Name: | **Abdullah** |
| Student ID number | **13115676** |
| Ethics Application Number | **P183326** |
| 1st Supervisor Name | **Amanda Brooks** |
| 2nd Supervisor Name |  |

**This form must be completed and included with your project submission to Turnitin (paste it at the beginning of your dissertation). Failure to append these declarations may result in your project being rejected for marking.**

Predicting Competitive Swimming Performance Using Machine Learning Models P183326



**Certificate of Ethical Approval**

|  |  |
| --- | --- |
| Applicant: | Muhammad Abdullah |
| Project Title: | Predicting Competitive Swimming Performance Using Machine Learning Models |

This is to certify that the above named applicant has completed the Coventry University Ethical

Approval process and their project has been confirmed and approved as Low Risk

Date of approval: 19 Dec 2024

Project Reference Number: P183326

Muhammad Abdullah (6000CEM) Page 1 19 Dec 2024

# Abstract

This project explores how machine learning can be used to analyse and predict performance in competitive swimming. The goal was to understand which models are most accurate at forecasting swim times and how athletes can be grouped based on their performance. After researching past studies and reviewing existing techniques, I prepared and cleaned a large swimming dataset to use in this analysis.

Different models were developed and tested, including Linear Regression, a Multilayer Perceptron (MLP), and XGBoost. XGBoost turned out to be the most effective, offering very accurate predictions and clear insights into what features were most important, such as race, distance and ranking. I also used a Random Forest Classifier to sort swimmers into performance categories, and K-Means Clustering to group them without using labels. These methods helped highlight trends and patterns in performance, which could be useful for coaches and analysts.

While advanced models like LSTM were considered for tracking performance over time, they were not implemented due to a lack of longitudinal data per swimmer. However, the project still shows that machine learning can be a powerful tool for sports analysis, and with further development, could support talent identification and personalized training. This work aims to show the value of using data to make better decisions in sports like swimming.

Contents

[Abstract 5](#_Toc195276056)

[1. Introduction 9](#_Toc195276057)

[2. Literature Review 10](#_Toc195276058)

[2.1 Introduction 10](#_Toc195276059)

[2.2 History of Sports Analytics and Its Evolution 10](#_Toc195276060)

[2.3 Traditional Methods in Sports Performance Analysis 11](#_Toc195276061)

[2.4 The Introduction of Technology in Sports Analytics 11](#_Toc195276062)

[2.5 The Role of Technology in Swimming Performance Analytics 12](#_Toc195276063)

[2.6 The Importance of Sports Analytics in Swimming Performance Prediction 12](#_Toc195276064)

[2.7 Research Objective: Enhancing Swimming Performance Prediction Using Machine Learning 13](#_Toc195276065)

[2.8 Models Used In Case Studies 13](#_Toc195276066)

[2.8.1 Linear and Advanced Regression Models 13](#_Toc195276067)

[2.8.2 Artificial Neural Networks 14](#_Toc195276068)

[2.8.3 Self-Organizing Maps and Clustering Techniques 14](#_Toc195276069)

[2.8.4 Classification Models and Ensemble Methods 15](#_Toc195276070)

[2.8.5 Linear Models: 15](#_Toc195276071)

[2.8.6 Yustres Amores et al. (2023): 15](#_Toc195276072)

[2.9 Predictive Models to be used for Swimming Performance Analysis In Project 16](#_Toc195276073)

[2.9.1 Regression Models for Performance Prediction 16](#_Toc195276074)

[2.9.2 Artificial Neural Networks for Nonlinear Prediction 16](#_Toc195276075)

[2.9.3 Decision Trees and Random Forest for Classification and Ranking 16](#_Toc195276076)

[2.9.4. Clustering Models for Performance Segmentation 17](#_Toc195276077)

[2.9.5 XGBoost for Enhanced Regression Performance and Feature Importance 17](#_Toc195276078)

[2.10 Gaps in Research and Opportunities for Future Study 18](#_Toc195276079)

[2.10.1 Limited Dataset Size and Heterogeneity 18](#_Toc195276080)

[2.10.2 Feature Selection and Data Preprocessing Challenges 18](#_Toc195276081)

[2.10.3 Performance Classification and Talent Identification 19](#_Toc195276082)

[2.10.4 Clustering for Performance Segmentation 19](#_Toc195276083)

[2.11 Practical Implications for Coaches and Athletes 20](#_Toc195276084)

[2.12 Summary 21](#_Toc195276085)

[3 Research Methodology 22](#_Toc195276086)

[3.1 Introduction 22](#_Toc195276087)

[3.2 Research Approach and Design 22](#_Toc195276088)

[3.3 Methods Considered 22](#_Toc195276089)

[3.4 Data Collection and Preprocessing 23](#_Toc195276090)

[3.5 Predictive Modeling Approaches 24](#_Toc195276091)

[3.6 Ethical Considerations 25](#_Toc195276092)

[3.7 Limitations and Challenges 25](#_Toc195276093)

[3.8 Conclusion 25](#_Toc195276094)

[4. Results 26](#_Toc195276095)

[4.1 Data Preprocessing and Feature Engineering 26](#_Toc195276096)

[4.2 Data Cleaning and Transformation 26](#_Toc195276097)

[4.2.1 Handling Missing Values 27](#_Toc195276098)

[4.2.2 Data Type Conversions 27](#_Toc195276099)

[4.3 Linear Regression Model 31](#_Toc195276100)

[4.4 Multilayer Perceptron Neural Network 36](#_Toc195276101)

[4.5 XGBoost Regression 42](#_Toc195276102)

[4.5.1 Per-Event Performance 43](#_Toc195276103)

[4.6 Random Forest Classifier 46](#_Toc195276104)

[4.6.1 Real-World Use of Random Forest Model 49](#_Toc195276105)

[4.7 Clustering Model 50](#_Toc195276106)

[4.7.1 Notable Swimmers Identified in High-Performance Groups 56](#_Toc195276107)

[5.0 Evaluation and Analysis 61](#_Toc195276108)

[5.1 Model Comparison and Insights 61](#_Toc195276109)

[5.2 Clustering and Classification 62](#_Toc195276110)

[5.3 Implications and Real-World Relevance 63](#_Toc195276111)

[5.4 Limitations 63](#_Toc195276112)

[5.5 Best Performing Model 63](#_Toc195276113)

[6. Project Management Approach 65](#_Toc195276114)

[6.1 Project Methodology 65](#_Toc195276115)

[6.2 Supervisor Feedback 67](#_Toc195276116)

[6.3 Gantt Chart Timeline 68](#_Toc195276117)

[6.4 Supervisor Meetings 69](#_Toc195276118)

[7. Conclusion 70](#_Toc195276119)

[7.2 Introduction 70](#_Toc195276120)

[7.3 Achievements 70](#_Toc195276121)

[7.4 Future Development 71](#_Toc195276122)

[8. Reflection 72](#_Toc195276123)

[9. References 73](#_Toc195276124)

[11. Appendix – Github Link 74](#_Toc195276125)

# Introduction

In competitive swimming, even a fraction of a second can make the difference between winning and losing. As the sport becomes more data-driven, coaches and analysts are increasingly turning to machine learning to better understand performance, identify talent, and predict race outcomes. While previous research has shown that historical swim data holds valuable insights, there is still a need for models that not only predict swim times accurately but also provide meaningful groupings and classifications of athletes.

The motivation behind this project stems not only from the technical challenges of building predictive models but also from a personal connection to the sport. Having been involved in swimming since childhood and developing a strong passion for it over the years, I have always been fascinated by the small margins that separate good performances from great ones. This personal experience inspired me to explore how data and machine learning could be used to better understand swimming performance, support talent development, and provide coaches with actionable insights.

This project aims to address these challenges by investigating whether machine learning can be used to accurately predict swim times and classify swimmers based on their competitive performance. The focus is on identifying the most effective models and understanding which features, such as race distance, event type, and swimmer ranking, have the strongest impact on outcomes. The study also explores clustering techniques to uncover natural performance groups without predefined labels.

While previous research has demonstrated that historical swimming data holds valuable information, there remains a gap in how machine learning is used to both predict race outcomes and provide meaningful athlete classifications. Many studies focus only on predicting race times without segmenting swimmers into performance categories, or they lack model transparency, limiting their usefulness for coaches and analysts.

# Literature Review

## 2.1 Introduction

In recent years, the integration of technology and data science into sports has significantly transformed how athlete performance is assessed, predicted, and optimized. In disciplines such as swimming, where outcomes are determined by fractions of a second, the ability to accurately analyze performance data has become increasingly valuable. Traditionally, performance analysis in swimming relied on observational methods, basic statistical evaluations, and physiological assessments. While these approaches provided useful insights, they could not often capture complex interactions between technical, physiological, and contextual variables.

With the advancement of machine learning and the growing availability of historical performance data, there has been a notable shift toward data-driven decision-making in sports science. Research has explored the use of models such as linear regression, artificial neural networks, and ensemble learning techniques to predict race outcomes and categorize athletes based on performance. However, several gaps remain in the literature, including limited model interpretability, lack of integration between classification and clustering approaches, and challenges in feature selection and data quality.

This literature review critically examines existing research related to machine learning in sports performance prediction, with a specific focus on swimming. It discusses traditional and modern analytic techniques, reviews the application of regression, classification, and clustering models, and identifies key opportunities for further exploration. Through this review, the foundation is established for addressing the central research question of this study, which is:

*Can historical swimming data predict race outcomes and identify key factors influencing performance in competitive swimming?*

## 2.2 History of Sports Analytics and Its Evolution

The application of analytics in sports has a long history, dating back to the early 20th century when teams and coaches began systematically recording performance metrics to gain a competitive advantage. One of the earliest known applications was in baseball, where statistical analysis of player performance improved scouting and game strategies (James, 1982). This movement gained widespread attention with the introduction of Sabermetrics, a data-driven approach that transformed player evaluation and team management in Major League Baseball (Lewis, 2003).

Basic descriptive statistics like player averages, win-loss records, and performance ratios were the foundation of sports analytics at first (Sharma et al., 2019). These techniques relied more on subjective assessments than on objective data collection, and they were mostly manual and observational. But as technology developed, analytics became a data-driven field that combined complex computer modeling with real-time data collecting to improve performance assessment and decision making.

## 2.3 Traditional Methods in Sports Performance Analysis

Before the emergence of advanced computing, coaches and analysts used manual scouting reports, video analysis, and basic statistical techniques to assess player and team performance. Some of the key traditional methods included:

* Time-motion analysis, which tracked an athlete’s movement and effort levels during a game (Barris & Button, 2008).
* Biomechanical assessments are used to evaluate the efficiency of movement and technique in sports like swimming, running, and weightlifting (Bartlett, 2006).
* Physiological monitoring, which analyses metrics such as heart rate, oxygen uptake (VO₂ max), and lactate threshold to assess an athlete’s conditioning (Pyne et al., 2004).

These approaches had serious drawbacks, even though they established the foundation for performance evaluation. Small sample sizes, human bias, and the incapacity to capture complex relationships between several performance indicators were the limitations of traditional approaches (Bishop, 2008). Due to these constraints, there was a need for analytics that were more accurate, scalable, and real-time; as a result, technology was included into sports performance evaluation.

## 2.4 The Introduction of Technology in Sports Analytics

The digital revolution of the late 20th and early 21st centuries marked a turning point in sports analytics. The introduction of wearable sensors, computer vision, and machine learning transformed how performance data was collected, analyzed, and applied in training and competition. Some of the most influential advancements include:

* Wearable Tracking Devices GPS systems and accelerometers enable real-time tracking of speed, movement patterns, and workload in endurance and team sports (Aughey, 2011).
* Computer Vision and Motion Capture AI-driven biomechanical assessments optimize movement efficiency and reduce injury risks (Glazier, 2010).
* Big Data and Machine Learning The integration of vast datasets allows for outcome prediction, talent scouting, and personalized training programs (Bunker & Thabtah, 2019).

These innovations have fundamentally changed the way athletes training, game strategy, and injury prevention, shifting decision-making from intuition-based approaches to evidence-driven methodologies.

## 2.5 The Role of Technology in Swimming Performance Analytics

Swimming has benefited significantly from developments in sports technology as it is a sport that is regulated by exact time measurements, stroke efficiency, and hydrodynamic characteristics. Swimming performance is mostly influenced by biomechanics, physiological fitness, and technical execution, in contrast to team sports where tactics and strategy are crucial (Toussaint & Beek, 1992). Predictive modeling, AI-based stroke analysis, and real-time tracking have all been used to help athletes and coaches maximize training schedules and increase race efficiency.

Several key technological applications have enhanced swimming analytics:

* Hydrodynamic analysis, utilizing underwater cameras and motion sensors to evaluate drag forces and propulsion efficiency (Puel et al., 2016).
* Stroke efficiency tracking, where AI-powered models assess stroke length, rate, and velocity to refine swimming technique (Felder et al., 2019).
* Real-time physiological monitoring, employing wearable sensors to track heart rate, oxygen consumption, and fatigue levels during training sessions (Costa et al., 2013).
* Predictive performance modeling, where historical data and machine learning algorithms forecast race outcomes and identify key performance factors (Maszczyk et al., 2012).

By integrating AI and data analytics, swimming has transitioned from traditional coaching methods to a more scientific and data-driven approach.

## 2.6 The Importance of Sports Analytics in Swimming Performance Prediction

The use of artificial intelligence (AI) and machine learning (ML) for predicting competitive swimming results has been made possible by the growing availability of previous performance data. To give more precise and accurate assessments of an athlete's potential, machine learning models examine a variety of physiological, biomechanical, and environmental factors (Allen et al., 2014).

Compared to traditional linear regression models, which rely on a limited set of predefined factors, machine learning-based approaches are capable of identifying complex, non-linear relationships between variables. For example, while conventional models might correlate race times with basic physiological metrics, Machine learning models can automatically detect hidden performance patterns, including subtle stroke inefficiencies, race pacing strategies, and the impact of turn speeds (Maszczyk et al., 2012).

Machine learning can play a vital role in identifying and developing potential in addition to predicting success. Machine learning models may identify young swimmers with promise and forecast their long-term potential by examining large databases of athlete performance patterns, training records, and physical characteristics. Instead of depending on subjective evaluations, national teams, and professional organizations may now objectively analyze talent because of this data-driven scouting strategy.

Machine Learning helps in customizing training programs by analyzing an athlete's strengths and weaknesses. AI-driven systems may optimize recovery times, modify training loads, and improve method tactics to enhance performance by utilizing real-time data. AI-based injury prevention models minimize the risk of overtraining and performance peaks by predicting injury risks and optimizing training intensities based on workload and biomechanical data (Neiva et al., 2017).

## 2.7 Research Objective: Enhancing Swimming Performance Prediction Using Machine Learning

While sports analytics has seen significant advancements, gaps remain in how machine learning is applied to swimming performance prediction. Instead of creating comprehensive models that include training load analysis, physiological adaptation over time, and stroke biomechanics, many current studies concentrate on single-race forecasts, so there are chances for additional innovation because swimming lacks explainable AI and real-time analytics.

This study aims to bridge these gaps by leveraging historical swimming data and machine learning models to improve race outcome forecasting and training optimization. The objective is to develop a comprehensive framework that:

* Accurately predicts race times and long-term performance trends.
* Identifies key determinants of success in competitive swimming.
* Provides data-driven training insights for coaches and athletes.

The next sections will explore specific ML techniques used in swimming analytics, the challenges in implementation, and the future direction of predictive performance modeling.

## 2.8 Models Used In Case Studies

### 2.8.1 Linear and Advanced Regression Models

Linear regression models have long been used to establish baseline relationships between input variables and race outcomes. Early studies demonstrated correlations between physiological metrics (e.g., VO₂ max, stroke rate) and performance times. However, linear models assume a constant rate of change between variables, a simplification that does not hold for the multifactorial nature of swimming. To capture non-linear relationships, researchers have applied methods such as polynomial regression and generalized additive models. For instance, Maszczyk et al. (2012) implemented traditional regression by incorporating logarithmic transformations and interaction terms to better model performance in 50m and 800m freestyle events. Their approach allowed them to capture the non-linear influence of factors like body size and stroke mechanics. Nevertheless, even these advanced regression models require prior assumptions about the functional form of relationships, limiting their flexibility when dealing with very high dimensional or complex data.

### 2.8.2 Artificial Neural Networks

Artificial neural networks have emerged as one of the most promising tools in predicting swimming performance because they can model complex, non-linear interactions without needing explicit assumptions about data distribution. In the study by Maszczyk et al. (2012), a cohort of competitive swimmers was assessed on various metrics such as lung capacity, strength (e.g., standing long jump), specific swimming technique measures (e.g., turn and glide times, stroke cycle characteristics), and anthropometric variables (e.g., hand length, body height).

The key findings that can be extracted are directly related to neural networks, specifically multilayer perceptron architectures, demonstrated lower absolute prediction errors compared to conventional regression models for both 50m and 800m freestyle events. The MLP model successfully integrated non-linear relationships among diverse variables that effectively captured the impact of subtle interactions between technical and physiological factors. The research was conducted for a 12-month training period, and separate testing groups were used (n = 60 swimmers) to compare predicted race times with actual performance outcomes. This two-stage approach validated the neural network’s predictive power.

### 2.8.3 Self-Organizing Maps and Clustering Techniques

Self-organizing maps are an unsupervised learning approach that reduces high-dimensional data to a lower-dimensional, typically two-dimensional grid while preserving the topological relationships among data points (Kohonen, 1995). This method has been used not only for performance prediction but also for clustering athletes with similar characteristics.  
Fidos-Czuba et al. (2015) implemented SOMs to cluster competitive swimmers based on their performance data, stroke styles, and physiological characteristics.

The research combined traditional regression models with neural network approaches, specifically Kohonen’s self-organizing maps, to classify swimmers based on performance and physiological profiles. Clustering was used to identify the talent. A visualization cluster on a two-dimensional grid allowed coaches to identify groups of swimmers with similar strengths and weaknesses easily. For example, one cluster might consist of swimmers with high stroke efficiency but lower explosive power, suggesting a need for targeted strength training. The SOMs allowed for the grouping of athletes into performance categories without relying on pre-labeled data, thereby supporting early talent identification initiatives. The study demonstrated that unsupervised clustering via SOMs could effectively identify groups of swimmers with similar performance characteristics, which is valuable for early talent identification and personalized training.

### 2.8.4 Classification Models and Ensemble Methods

Classification models, such as Support Vector Machines (SVMs) and gradient boosting algorithms (e.g., XGBoost), have also been employed to classify swimmers into performance tiers (Zadeh, 2002). These methods are particularly useful in situations where the goal is to predict discrete outcomes, such as whether a swimmer will finish in the top three positions at a competition.

Studies have shown that ensemble methods, which combine multiple models, can effectively classify athletes based on historical data. For instance, gradient boosting methods have been successful in determining the probability of a swimmer achieving a podium finish by analyzing past race outcomes, physiological data, and even training load metrics.

In practice, these classification models use a combination of cross-validation techniques and hyperparameter tuning to optimize performance. Although many studies focus on regression for time prediction, recent works are increasingly exploring ensemble methods to classify performance levels.

### 2.8.5 Linear Models:

This study analyzed career performance trajectories of Olympic swimmers using individual quadratic trends. By applying mixed linear models and adjusting for factors like the Olympic year effect and technological advancements (e.g., the introduction of polyurethane swimsuits), the researchers established benchmarks for talent development.

They found that male swimmers typically reach peak performance around 24.2 years, while females peak at about 22.5 years. Such benchmarks help in evaluating the progress of young athletes relative to elite performance standards.

### 2.8.6 Yustres Amores et al. (2023):

Using a large dataset from international swimming competitions (FINA World Championships), this study modeled performance differences by continent and examined the impact of early specialization. The authors applied statistical methods such as regression, ANOVA, and ANCOVA to compare performance across different groups.

The study found that swimmers with junior competition experience tend to perform better in senior competitions, and that there are significant performance differences by continent. For example, European and Asian swimmers often showed distinct trajectories compared to their American counterparts.

## 2.9 Predictive Models to be used for Swimming Performance Analysis In Project

### 2.9.1 Regression Models for Performance Prediction

Linear and nonlinear regression models have been widely applied in sports science to predict swimming times based on historical data. Maszczyk et al. (2012)​maszczyk-et-al-2012-app…demonstrated the effectiveness of regression in predicting performance in freestyle events. In this dataset, a Multiple Linear Regression model could be used to predict a swimmer’s final time based on variables such as distance, ranking, and event type.

### 2.9.2 Artificial Neural Networks for Nonlinear Prediction

Artificial Neural Networks (ANNs) are effective in capturing nonlinear patterns in sports data, particularly when there are multiple interacting factors influencing performance. Research by Maszczyk et al. (2012)​maszczyk-et-al-2012-app…showed that neural networks outperform traditional regression models in predicting 50m and 800m freestyle times. Given this dataset, a Multilayer Perceptron (MLP) neural network could be implemented using input nodes such as event type, distance, ranking, and milliseconds, with the output layer predicting future swim times. This approach allows for better adaptability when modeling how different race conditions influence performance.

For example, an 8-4-1 neural network (8 input features, 4 hidden neurons, 1 output prediction) could be structured as follows:

* Input Layer: Event, Distance, Sex, Ranking, Nationality, Meet, Time in Milliseconds, Date
* Hidden Layer: Captures complex interactions like the effect of race strategy or competition level
* Output Layer: Predicted Swim Time

This approach provides higher accuracy in forecasting compared to linear models, especially when swimmer performance is influenced by nonlinear factors.

### 2.9.3 Decision Trees and Random Forest for Classification and Ranking

Decision trees and ensemble methods such as Random Forests provide interpretable models that classify swimmers into different performance levels based on ranking and historical swim times. These models are particularly useful for talent identification and predicting the likelihood of a swimmer reaching the finals based on past performances.

For instance, using a Decision Tree Classifier, swimmers could be grouped into "High Performance" and "Developing Athletes" based on:

* If ranking ≤ 3, classify as "Elite."
* If ranking between 4 and 8, classify as "Competitive."
* If ranking > 8, classify as "Developing."

A Random Forest model could further improve this classification by aggregating multiple decision trees to minimize overfitting. This model could be implemented using swimmer’s ranking, nationality, and past swim times as input features to determine final classification.

### 2.9.4. Clustering Models for Performance Segmentation

Clustering algorithms, such as K-Means Clustering, can be used to segment swimmers into performance groups based on their attributes. For example, using Time in Milliseconds, Distance, and Meet Type, the model could group athletes into:

* Cluster 1: World-class performers
* Cluster 2: National-level competitors
* Cluster 3: Regional competitors

This method, supported by Wilk et al. (2015)​12\_fidos-czuba.o-kozlow,helps in identifying athlete development trajectories, which is crucial for coaching and performance analysis.

### 2.9.5 XGBoost for Enhanced Regression Performance and Feature Importance

Extreme Gradient Boosting (XGBoost) is a powerful ensemble learning method that builds a series of decision trees using gradient boosting. It is well-known for its high predictive accuracy, efficiency, and built-in feature importance analysis. XGBoost is particularly suited for handling structured data like race times, rankings, and categorical variables (e.g., event type, sex).

In swimming analytics, XGBoost can be used to predict final swim times by learning complex, nonlinear relationships between input features. Additionally, the model can highlight which variables—such as distance, sex, or historical ranking—have the greatest impact on performance. This dual benefit of prediction and interpretation makes XGBoost a valuable tool for coaches and analysts.

For instance, by inputting features like Event, Distance, Sex, Ranking, and Meet Type, the XGBoost model can:

* Accurately predict swim times across various race types
* Identify dominant factors influencing performance (e.g., distance might outweigh ranking for longer events)
* Enable coaches to adjust training based on data-driven insights

Due to its flexibility and scalability, XGBoost is increasingly being used in performance modeling and is particularly effective in datasets with mixed feature types and interactions.

## 2.10 Gaps in Research and Opportunities for Future Study

While significant advancements have been made in the application of machine learning to swimming performance prediction, several gaps remain in the literature. These gaps directly impact the ability to develop accurate, interpretable, and generalizable models for predicting swim times and classifying athletes into performance categories. Addressing these limitations is essential to improving race forecasting, athlete performance, and data-driven coaching strategies.

### 2.10.1 Limited Dataset Size and Heterogeneity

One of the primary challenges in swimming performance prediction is the availability of large, diverse, and representative datasets. Many prior studies, such as Maszczyk et al. (2012), have relied on small, homogenous datasets, often restricted to specific competitions, training groups, or national teams. Allen et al. (2014) also highlight that performance benchmarks are often based on data from a limited competitive context, making them less applicable to diverse swimming populations.This limitation reduces the generalizability of machine learning models, making them less effective in predicting performance across different swimming populations.

This project addresses this limitation by using a larger and more diverse dataset, capturing swim performances across multiple events and competitions, and instead of depending just on samples from a single season, the models may examine performance patterns over time by utilizing a longitudinal dataset that spans several seasons. The use of the Interquartile Range (IQR) approach for outlier removal, guarantees that inaccurate or extreme records do not skew model predictions.

### 2.10.2 Feature Selection and Data Preprocessing Challenges

A critical challenge in swimming analytics is determining which features most accurately predict performance. Previous studies have primarily relied on basic physiological and race-related metrics, such as stroke rate, body composition, and ranking, without fully exploring advanced feature selection techniques. While leaving out significant predictors might reduce model accuracy, choosing too many unimportant characteristics. As highlighted by Yustres et al. (2023), feature selection plays a crucial role in performance modeling, particularly in ensuring that the predictors used are meaningful across different swimmer types. Dutt-Mazumder et al. (2011) observed that many traditional models fail to capture complex, nonlinear relationships due to simplistic or manual feature inclusion strategies.

To address this, this study employs automated feature selection techniques, including XGBoost feature importance rankings, to identify the most significant predictors of swim times. Linear Regression serves as a baseline model to assess whether relationships between input variables and swim times are linear or require more complex nonlinear models such as MLP Neural Networks and XGBoost Regression.

By combining traditional regression techniques with advanced machine learning approaches, this project ensures that the most influential performance factors such as distance, ranking, and swimmer classification are effectively utilized in predictive modeling.

### 2.10.3 Performance Classification and Talent Identification

Beyond predicting race times, classifying swimmers into performance categories is crucial for talent identification and athlete development. Many previous studies have relied on simplistic classification methods, often using ranking alone as a performance indicator. However, ranking alone does not capture relative performance within an event, nor does it account for different competitive contexts. Traditional talent identification models often assume static ability, failing to capture the variability of performance across time and context (Abbott & Collins, 2004). By contrast, percentile-based classifications, as proposed by Yustres et al. (2019), offer a more context-sensitive method for athlete segmentation based on relative performance.

To address this limitation, this project implements a Random Forest Classification Model to categorize swimmers into Elite, Competitive, and Developing groups based on percentile rankings within each event. Unlike traditional classification models that rely only on ranking, this approach considers swim times in relation to the event’s overall distribution, ensuring a more fair classification.

By using Random Forest, a highly interpretable classification model, this study ensures that performance is robust, helping coaches identify promising athletes and tailor training programs accordingly.

### 2.10.4 Clustering for Performance Segmentation

While classification models segment athletes based on predefined categories, clustering techniques provide an unsupervised approach to group swimmers based on performance patterns. Prior studies have focused primarily on predictive modeling, with limited research on how clustering can enhance performance analysis and training optimization. Clustering techniques, such as Self-Organizing Maps, have been used in performance profiling (Maszczyk et al., 2012; Roczniok et al., 2007), but they are often applied in isolation, without integration into predictive frameworks. This limits their strategic value for forward-looking applications like training prioritization or performance forecasting.

This study employs K-Means Clustering to group swimmers based on pace (seconds per meter) rather than simply using race times. By normalizing swim times across distances, this clustering method allows for a fair comparison of performance across different events.

By implementing K-Means alongside classification models, this project introduces a hybrid approach where:

* Clustering identifies natural performance groups, helping to refine talent hunting and training programs.
* Classification models provide structured labels for swimmer development, supporting targeted coaching interventions.

This integration enhances the overall effectiveness of machine learning in performance analytics, offering both data-driven talent identification and predictive modeling for race outcomes.

## 2.11 Practical Implications for Coaches and Athletes

The integration of machine learning into swimming analytics has direct benefits for coaches, athletes, and sports organizations. This study facilitates data-driven decision-making in strategic planning, training, and talent development by utilizing clustering, classification, and predictive modeling approaches.

One of the primary benefits is personalized training optimization. If the models indicate that a swimmer’s performance is highly dependent on factors such as distance and event type, training programs can be tailored to focus on stroke-specific improvements and pacing strategies. This allows coaches to allocate training time more efficiently, prioritizing the most impactful performance factors.

Talent identification is enhanced through performance-based clustering and classification models. This research offers a more complex and data-driven monitoring method by grouping swimmers according to relative performance rather than absolute ranking. This helps sports organizations identify high-potential athletes early in their development, enabling more targeted investment in training and resources.

Machine learning-based benchmarking allows national teams to compare their athletes to international competitors. This can inform strategic decisions in team selection, resource allocation, and long-term athlete development, ensuring that data-driven insights translate into visible performance improvements.

By integrating multiple machine learning approaches, this project offers a comprehensive framework for swimming analytics, combining predictive modeling, classification, and clustering to provide actionable insights for performance enhancement.

## 2.12 Summary

The reviewed literature demonstrates that historical swimming data hold considerable potential for predicting race outcomes and identifying key performance indicators. While traditional models, such as linear regression, offer a baseline for prediction, they often fail to capture the non-linear and multifactorial nature of competitive swimming. Studies like Maszczyk et al. (2012) highlight the value of neural networks in improving predictive accuracy, while Allen et al. (2014) provide performance benchmarks critical for talent development, and clustering methods, as discussed by Fidos-Czuba et al. (2015), show promise in grouping athletes based on performance patterns.

Despite these advances, several gaps persist. Many studies suffer from limited or homogeneous datasets, insufficient feature diversity, and a lack of model interpretability. The integration of explainable AI remains underexplored, and clustering is often treated independently from predictive modeling. These limitations underscore the need for more comprehensive and transparent approaches in swimming analytics.

In this context, machine learning techniques such as linear regression, MLP neural networks, XGBoost regression, and random forest classification offer a powerful framework for enhancing performance prediction, athlete performance, and talent identification. The addition of K-Means clustering provides an unsupervised perspective, identifying natural groupings in swimmer performance based on pace. By combining these models, future systems can offer comprehensive profiling tools that assist coaches, analysts, and organizations in making data informed decisions in competitive swimming.

# 3 Research Methodology

## 3.1 Introduction

This section presents the methodology adopted to explore whether historical swimming data can predict race outcomes and identify key performance-influencing factors in competitive swimming. A structured approach ensures a thorough investigation, considering multiple methodological options before selecting the most suitable techniques. The study follows a data-driven approach, leveraging statistical and machine learning methods to develop predictive models based on historical performance records. The methodology focuses on selecting tools that provide reliable, interpretable, and generalizable insights while addressing data preprocessing, modeling, and validation challenges.

## 3.2 Research Approach and Design

The research follows a quantitative approach, as it relies on numerical data and computational models to derive insights. The design is predictive and analytical, seeking to uncover relationships between swimmer attributes and race performance. The study involves supervised learning models for time prediction and classification, as well as exploratory data analysis to understand the distribution and trends in the dataset. The dataset comprises historical swimming records, including attributes such as event type, distance, swimmer gender, ranking, and recorded race times. These attributes serve as input variables in the predictive models.

Given the nature of the research question, a predictive modeling approach is employed to estimate race times and classify swimmers into performance categories. Various machine learning and statistical methods are considered, ensuring that the chosen models align with the study's objectives. The selection process involves assessing the interpretability, accuracy, and computational efficiency of different techniques before finalizing the models.

## 3.3 Methods Considered

Before selecting the final methods, several approaches were evaluated to determine their suitability for predicting swim times and classifying swimmer performance. Initially, linear regression was considered as a baseline model due to its simplicity and interpretability. It provides insight into the relationship between independent variables (such as distance and ranking) and swim time. However, linear regression assumes a linear relationship between variables, which may not fully capture the complexity of swimming performance.

To address potential nonlinear relationships, a multilayer perceptron (MLP) neural network was explored. Neural networks have the advantage of learning complex patterns and interactions between variables, potentially improving predictive accuracy. However, initial attempts showed issues with convergence, requiring adjustments to hyperparameters such as the number of hidden layers, learning rate, and stopping criteria. Additionally, the computational intensity of neural networks made them more resource-demanding than traditional regression models.

For performance classification, random forest classification was selected after considering multiple alternatives, including logistic regression and support vector machines (SVM). Logistic regression was rejected due to its limited ability to handle complex feature interactions, while SVM was also evaluated but required extensive computational resources and parameter tuning, making them less practical than random forests. Random forest was chosen for its robustness, ability to handle both numerical and categorical variables, and resistance to overfitting. This model was used to classify swimmers into three performance categories Elite, Competitive, and Developing based on their percentile ranks in event-specific swim times.

To explore clustering techniques, K-Means clustering was initially considered to identify natural groupings of swimmers based on performance. However, it was found to be less effective due to the high variance in swim times across different events, which resulted in unstable cluster assignments. Consequently, supervised classification models were preferred for performance segmentation.

The final selection of methods was based on a balance between accuracy, interpretability, and computational feasibility. Linear regression provided a transparent baseline, offering a simple and interpretable model for performance prediction. MLP neural networks were selected for their ability to capture nonlinear relationships, significantly improving prediction accuracy. Random forest classification emerged as the most suitable method for categorizing swimmer performance, outperforming logistic regression and clustering alternatives.

## 3.4 Data Collection and Preprocessing

The dataset used in this study was selected due to its comprehensive coverage of historical competitive swimming performances, providing key variables such as event type, distance, swimmer gender, ranking, and recorded race times. The primary reason for choosing this dataset was its richness in performance-related metrics, allowing for an in-depth analysis of patterns in swimming times and rankings across various competitions. To effectively model race outcomes, it was crucial to have a dataset that contained both numerical and categorical features, enabling the application of predictive models that leverage structured data.

The dataset includes a diverse range of events and distances, which is essential for understanding performance variations across different swimming disciplines. By utilizing this dataset, the study aims to uncover significant predictors of race performance, evaluate the impact of different factors on swim times, and build robust predictive models that can forecast future race outcomes. The availability of time-series elements, such as event dates and competition history, also provides opportunities for trend analysis, making the dataset an ideal choice for answering the research question of how historical data can be used to predict and assess competitive swimming performance.

The dataset used in this study consists of historical swimming performance records containing essential attributes such as event type, distance, swimmer gender, ranking, and recorded race times. Since predictive models rely on high-quality input data, preprocessing steps were applied to clean and transform the dataset. Missing values were handled by replacing categorical gaps with the placeholder "Unknown," ensuring that incomplete records were not discarded unnecessarily.

Several transformations were applied to convert data into a format suitable for machine learning models. Swim times, originally recorded as strings (e.g., "00:23.86"), were converted into numerical values in seconds. Similarly, ranking values were transformed into a numeric format to facilitate mathematical operations. The dataset also contained outliers, which were identified and removed using the Interquartile Range (IQR) method. This approach ensured that extreme or erroneous values did not distort model training, improving overall accuracy.

Feature engineering played a critical role in refining the dataset for predictive modeling. Numeric variables such as race, distance and ranking were standardized using StandardScaler, while categorical variables such as swimmer gender and event type were encoded using OneHotEncoder. These transformations ensured that all features were properly formatted for machine learning algorithms.

## 3.5 Predictive Modeling Approaches

The study employed two key predictive models linear regression and MLP neural networks for time prediction and random forest classification for performance segmentation. Linear regression was implemented first as a benchmark model, providing a straightforward interpretation of feature importance. It achieved an RMSE of approximately 40.63 seconds, which, while informative, indicated room for improvement in capturing complex relationships.

To enhance predictive performance, an MLP neural network was developed with two hidden layers, optimizing its architecture through hyperparameter tuning. The final configuration included an (8,4) neuron structure, an adaptive learning rate, and early stopping to prevent overfitting. This model significantly improved accuracy, achieving an RMSE of 21.11 seconds and an R² score of 0.99165, demonstrating strong predictive power.

For classification, random forest classifiers were employed to segment swimmers into Elite, Competitive, and Developing categories. This model was selected over logistic regression due to its ability to handle nonlinear feature interactions and its superior performance in cross-validation. The random forest classifier achieved an accuracy of 95.7%, effectively distinguishing performance tiers. However, minor misclassification issues in the Elite category suggested potential refinements in feature selection.

While clustering techniques were initially explored, they were found to be less effective in producing meaningful groupings due to high intra-event variability. Instead, the supervised classification provided a more structured and interpretable approach to performance segmentation.

## 3.6 Ethical Considerations

The dataset used in this study does not contain personally identifiable information, ensuring compliance with ethical research standards. Care was taken to handle missing data responsibly and avoid biases in model training. Additionally, transparency was maintained by documenting preprocessing steps and model training procedures, allowing for reproducibility.

## 3.7 Limitations and Challenges

Despite the effectiveness of the selected models, certain challenges were encountered. The MLP neural network exhibited greater variance in cross-validation, indicating sensitivity to training conditions. This was mitigated by using regularization techniques and dropout layers to improve generalability. The random forest classifier occasionally misclassified elite swimmers, suggesting that additional performance-related features could improve classification accuracy.

Another limitation was the reliance on historical race data, which may not account for contextual factors such as training conditions or psychological variables influencing performance. Future work could explore integrating time-series forecasting techniques to assess performance trends over multiple seasons.

## 3.8 Conclusion

This research methodology systematically evaluates and justifies the selection of predictive models for analyzing historical swimming data. The study identifies the most effective approaches for time prediction and performance classification through a rigorous process of method evaluation, preprocessing, and modeling. The methodology ensures a balance of computational efficiency, and practical applicability, laying the foundation for future research in sports analytics and athlete performance modeling.

# 4. Results

## 4.1 Data Preprocessing and Feature Engineering

Before building any predictive models, it is important to prepare the dataset properly. This involves collecting relevant data, cleaning it, transforming various fields, and ensuring that all values are in a format suitable for machine learning models.

The dataset used in this project contains detailed records of swimming performances across various events and competitions. Each row in the dataset represents an individual swimmer's race, with columns capturing both event-related and swimmer-specific information.

Dataset Overview:

* **Event:** Describes the type of swimming event (e.g., 50m freestyle, 100m butterfly)
* **Distance:** The official race distance in meters
* **Sex:** The swimmer’s gender (Male or Female)
* **Ranking:** The swimmer’s position at the end of the race
* **Time 1 & Milliseconds:** Swim completion time in both string format (e.g., “00:24.78”) and numerical format (milliseconds)
* **Additional Fields:** Includes information like the race date (Date), location (Meet, Location), and whether the event was part of a relay

This dataset provided a strong foundation for analyzing and predicting race outcomes using various machine learning models as the data set included a variety of events of swimming races that provided the chance to apply multiple models and cover multiple patterns across different events

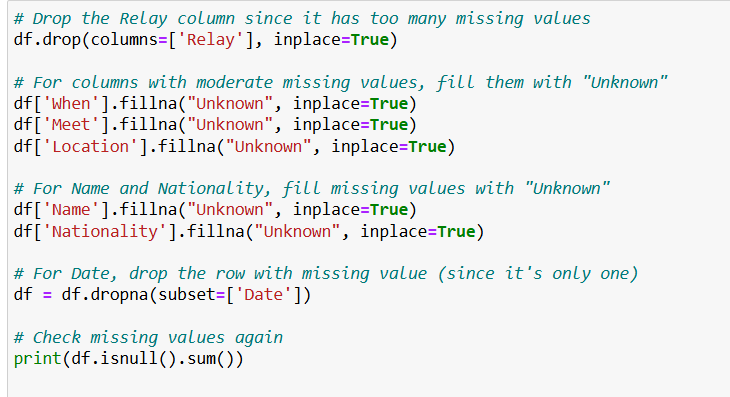
## 4.2 Data Cleaning and Transformation

Once the dataset was collected, several cleaning and transformation steps were performed using the pandas library to prepare the data for machine learning. Without this preparation, machine learning models could produce inaccurate results.

### 4.2.1 Handling Missing Values

Some columns contained missing entries, especially in non-essential fields like Relay, When, Meet, and Location. Instead of dropping entire rows, which could lead to a significant loss of data, we used the fillna() function to replace missing categorical values with the label "Unknown". This allowed us to preserve key race records while maintaining consistency.

In contrast, the Relay column, which had a high number of missing values and limited relevance to our modeling task, was removed entirely using drop().



### 4.2.2 Data Type Conversions

* **Time Conversion:**

The swim time was originally recorded as a string in the Time 1 column, using formats like "00:23.86". These strings were converted into numeric values in seconds using a custom function. This new numerical column, Time\_seconds, became the main target for all prediction models.



* **Date Conversion:**

To allow for future time-based analysis, the Date column was converted into a datetime format using pd.to\_datetime():

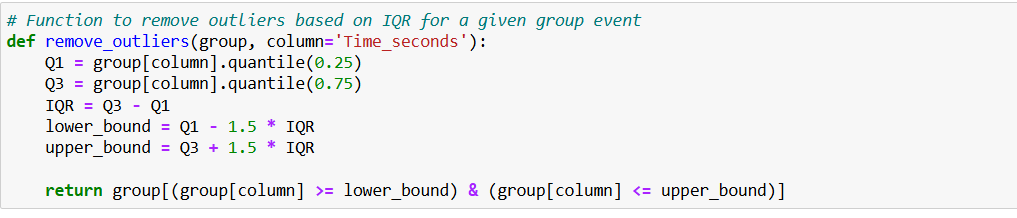
One record with an unreadable date was dropped using dropna() on the Date column.

* **Ranking Conversion:**

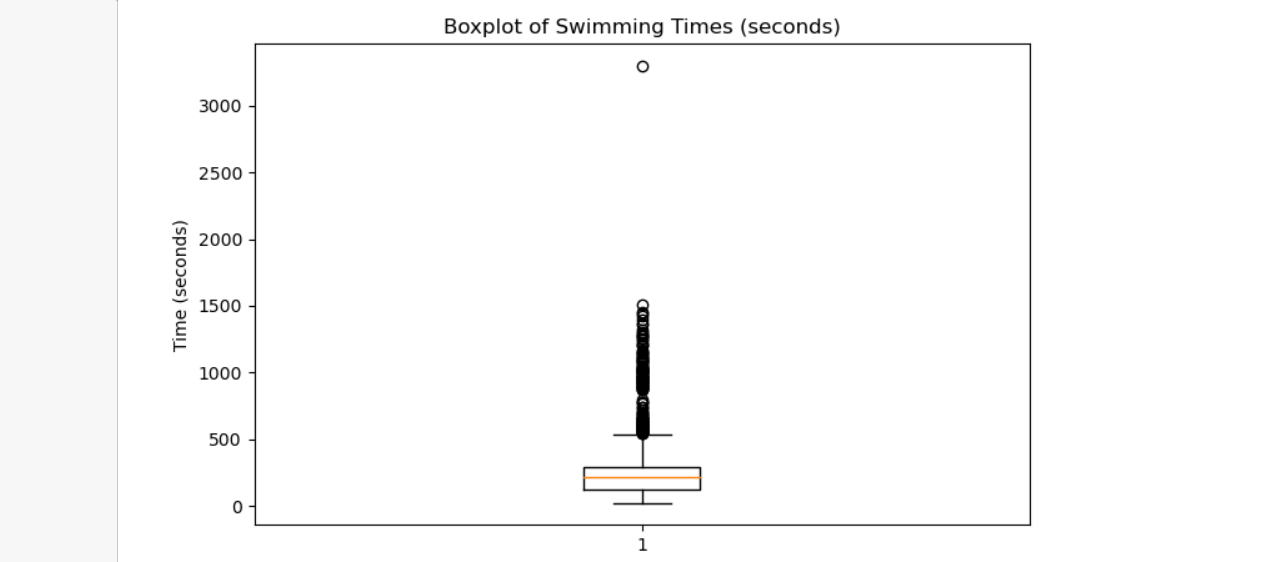
The Ranking field contained a mix of numbers and non-numeric values. These were cleaned and converted to a new column called Ranking\_numeric, which stores only numerical values and is suitable for use in modeling. (If needed, non-numeric rankings like "DSQ" or "DNF" would typically be converted to NaN and handled accordingly though this wasn’t shown in the snippet).

Outliers are data points that deviate significantly from the rest of the dataset. In swimming data, this could be due to recording errors, disqualifications, or extreme underperformance. To remove these anomalies:

* We applied the Interquartile Range (IQR) method on a per-event basis.
* For each event, only swim times within the range [Q1 − 1.5 × IQR, Q3 + 1.5 × IQR] were retained.
* This ensured that the model was trained on realistic and typical performance data, improving both stability and accuracy.

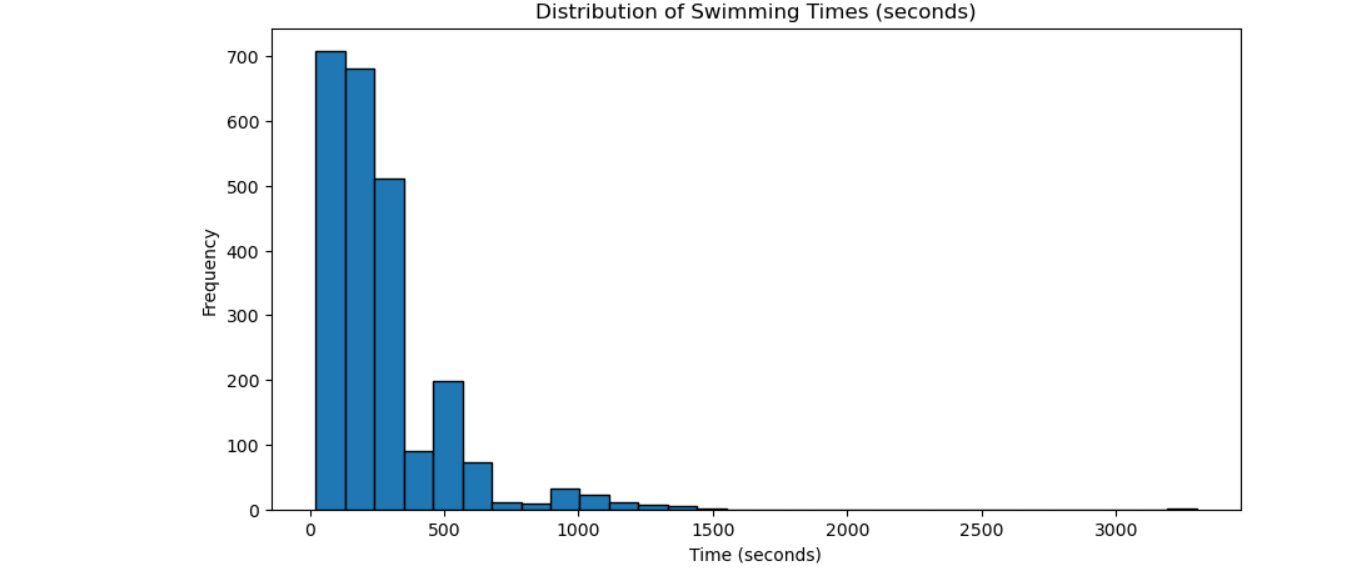


The code snippet used to remove outliers



**Figure – Box Plot Represents Outliers**

As we can see from the above boxplot, it provides a visual summary of all swimming times in the dataset, measured in seconds. It highlights the presence of outlier swim times that are significantly higher than the majority. Most race times fall within a compact range near the bottom of the graph, with the interquartile range (IQR) tightly clustered below 400 seconds. However, a large number of data points beyond the upper whisker indicate extreme values, some even exceeding 3000 seconds. These could result from timing errors, incomplete races, or unusually long-distance events, and they have the potential to skew model training if not removed. This boxplot influences the decision to apply interquartile range based outlier removal during data cleaning, ensuring that the predictive models would focus on accurate swim performances.



**Figure - Histogram of Swimming Time Distribution**

The histogram illustrates the distribution of swim times in seconds across the dataset. The majority of swim times are clustered at the lower end of the x-axis, with the highest frequency between 0 and 300 seconds. This indicates that most events are short to mid-distance races such as 50m, 100m, or 200m, which typically fall within this time range. As the swim time increases, the frequency of observations drops sharply, resulting in a right-skewed distribution. A long tail can be seen extending toward higher values, including some events above 1000 seconds, and even a few outliers beyond 3000 seconds. These extreme values are rare and likely correspond to either long-distance events, such as 1500m freestyle or possible data entry anomalies. This distribution confirmed the need for outlier removal, as the skew could negatively impact model training and reduce prediction accuracy for typical performance ranges.

After this preprocessing, the dataset was cleaner, more consistent, and ready to be used in building reliable predictive models.

The dataset was further prepared for machine learning by selecting key features and transforming them into formats suitable for predictive modeling. The process ensured that the data is consistent, numerically meaningful, and can be used by various types of models.

The following four features are based on their relevance to predicting swim performance:

* **Distance** (numeric): Represents how long the race is in meters.
* **Sex** (categorical): Indicates whether the swimmer is male or female.
* **Event** (categorical): The type of swimming event (e.g., 100m butterfly, 50m freestyle).
* **Ranking\_numeric** (numeric): A numerical form of the swimmer's final placement in the event.

These features were chosen because they directly influence a swimmer’s race time. For example, longer distances naturally result in longer times, while sex and event type affect performance due to physiological and technical differences. A swimmer's ranking is also closely related to their speed within the competition. To make the selected features usable by machine learning models, we applied the following preprocessing techniques:

* **Standardization:**  
  All numerical features (Distance and Ranking\_numeric) were standardized using StandardScaler. This transforms the values to have a mean of 0 and a standard deviation of 1. Standardization ensures that features with different ranges (e.g., 50m vs. 1500m) don’t unfairly dominate the learning process.
* **Encoding Categorical Variables:**  
  The categorical variables Sex and Event were transformed using OneHotEncoder. This technique creates binary (0 or 1) columns for each unique category, making them interpretable by machine learning models without introducing bias in the form of numerical magnitude.

All preprocessing steps were built into pipelines, which ensured that the same transformations were consistently applied during both model training and testing. These pipelines were reused across all predictive models for fairness and efficiency.

## 4.3 Linear Regression Model

Linear Regression was selected as our baseline model for predicting swimming performance. It is one of the simplest and most interpretable algorithms in machine learning, assuming a direct, straight-line relationship between input features and the target variable. While not designed to handle complex interactions or non-linear patterns, it is an essential starting point for evaluating how well basic statistical patterns can explain race outcomes.

The Linear Regression model was trained using the cleaned and preprocessed dataset, where swim time in seconds was the target variable. All input features had been scaled or encoded as described in the previous section.

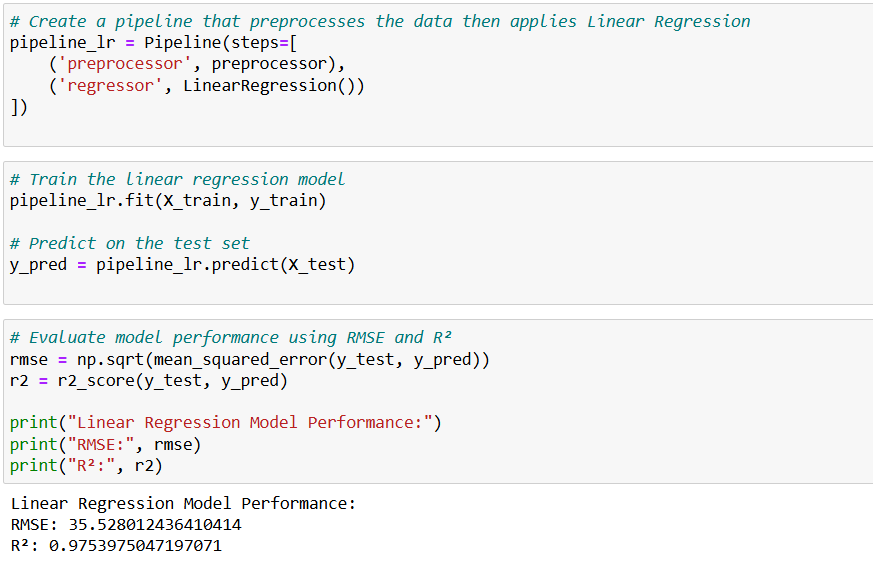
The Linear Regression model was implemented using the LinearRegression() class from the sklearn.linear\_model module. Preprocessing and model training were streamlined using the Pipeline and ColumnTransformer tools from scikit-learn. This ensured consistent transformations and easier model evaluation.

* Numerical features (Ranking\_numeric, Distance) were scaled using StandardScaler()
* Categorical features (Sex, Event) were encoded using OneHotEncoder(handle\_unknown='ignore')
* Data was split using train\_test\_split() into 80% training and 20% testing sets
* The model was evaluated using:

Root Mean Squared Error (RMSE) via mean\_squared\_error()

R² Score via r2\_score()

All preprocessing and model steps were wrapped in a single pipeline object, ensuring clean and reproducible experimentation.



We constructed a machine learning pipeline that combined data preprocessing with the model training step. This pipeline included:

* The transformations applied to the input features (StandardScaler, OneHotEncoder)
* A LinearRegression() estimator to perform the actual prediction

The model was trained using 5-fold cross-validation, which divides the data into five parts and rotates the training and validation sets. This method ensures more reliable performance estimates and helps prevent overfitting to a single split of the data.

The model’s goal was to predict the Time\_seconds column — the duration it took each swimmer to finish their race.

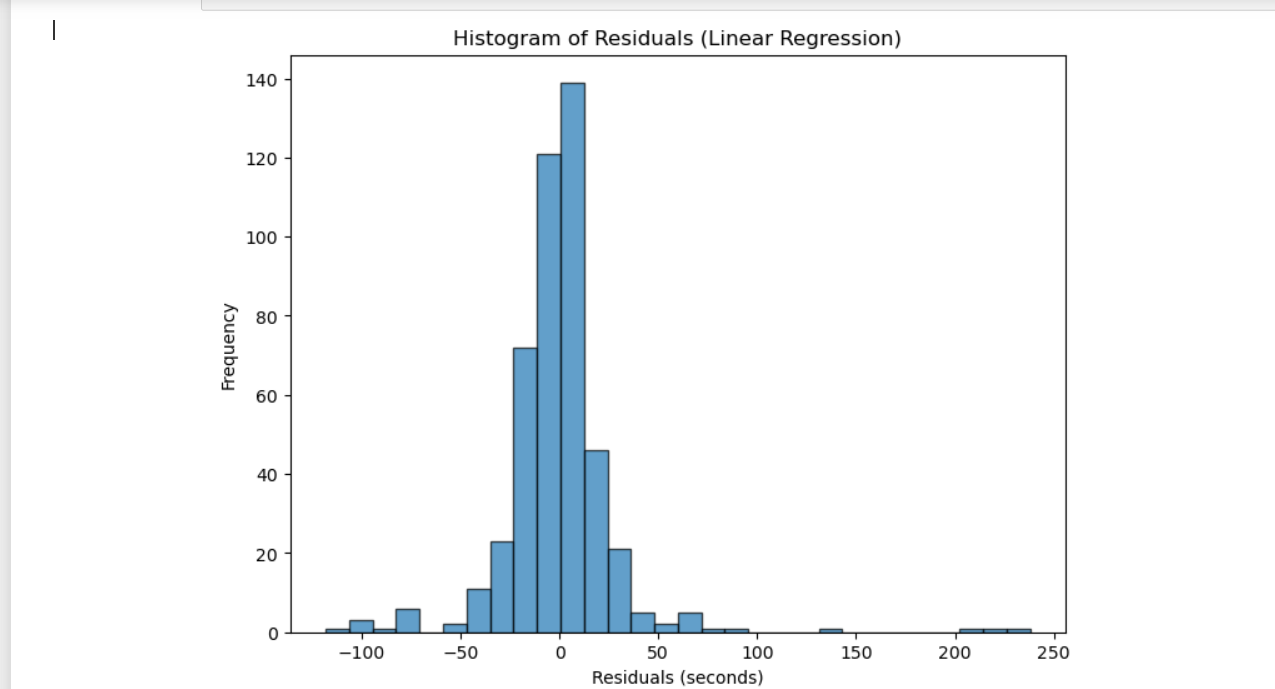
After training and validating the model, the following performance results were obtained:

* **Root Mean Squared Error :**  
  The average prediction error was approximately 40.63 seconds, with a standard deviation of ±26.57 seconds.  
  RMSE tells us how far off our model’s predictions are from the actual values on average. An error of 40 seconds is very large, especially in short-distance events like 50m or 100m freestyle, where races are often completed in less than a minute. This means that the model's predictions were not reliable for short or even moderate races.
* **R² Score:**  
  The model showed a moderate R² score, which means it could explain part, but not all, of the variance in swim times.  
  A high R² would indicate that the model can accurately capture how swim time changes based on the input features. A moderate value suggests that the Linear Regression model missed many important patterns, likely due to the complex and nonlinear nature of athletic performance.

While Linear Regression offered a useful starting point, it was not sufficient for accurate swim time prediction. The high RMSE value showed that the model struggled to make precise predictions, especially in events where seconds are crucial. Additionally, the moderate R² score confirmed that simple linear relationships were not enough to model the complexity of swimming performance, which can be influenced by non-linear interactions between event type, gender, distance, and swimmer rank.

These limitations motivated the shift toward more advanced models, like neural networks and ensemble methods, which are capable of learning from complex patterns in data.

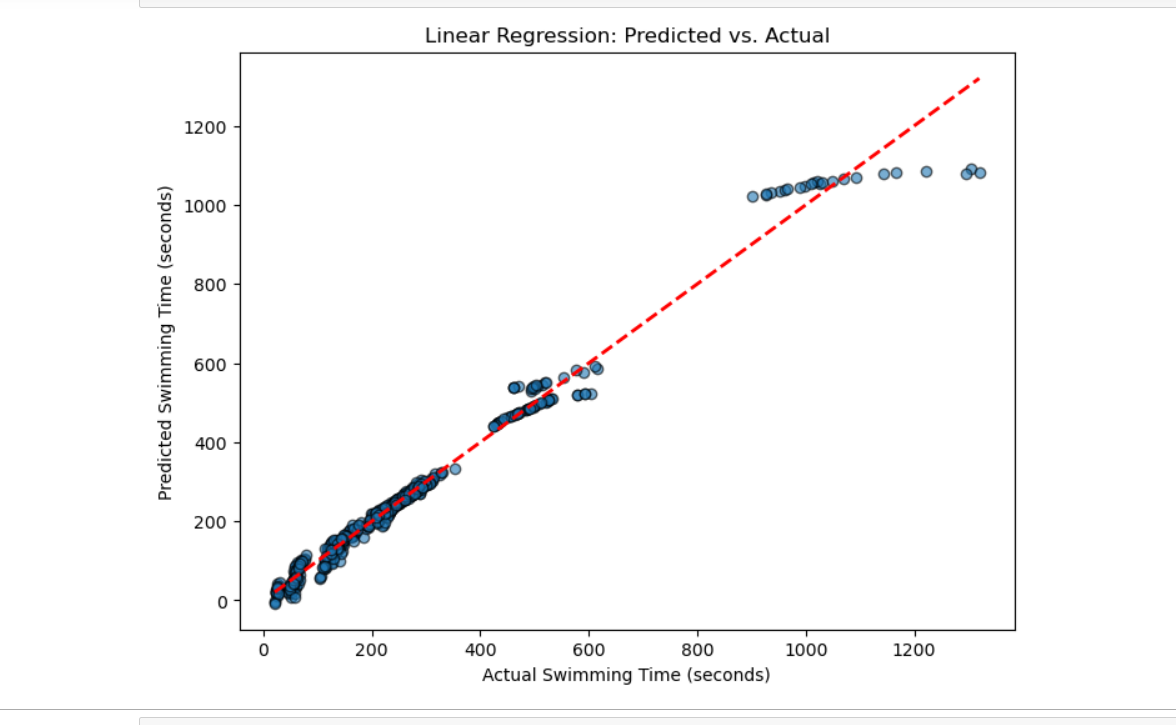
Some data visualisations were made for a better understanding of the model, which are below:



**Figure - Histogram of Residuals from the Linear Regression Model**

The histogram of residuals shows the distribution of errors, i.e., how far off the predictions were from the actual values. Ideally, it was expected to be a normal distribution centered at zero, which would suggest unbiased predictions.

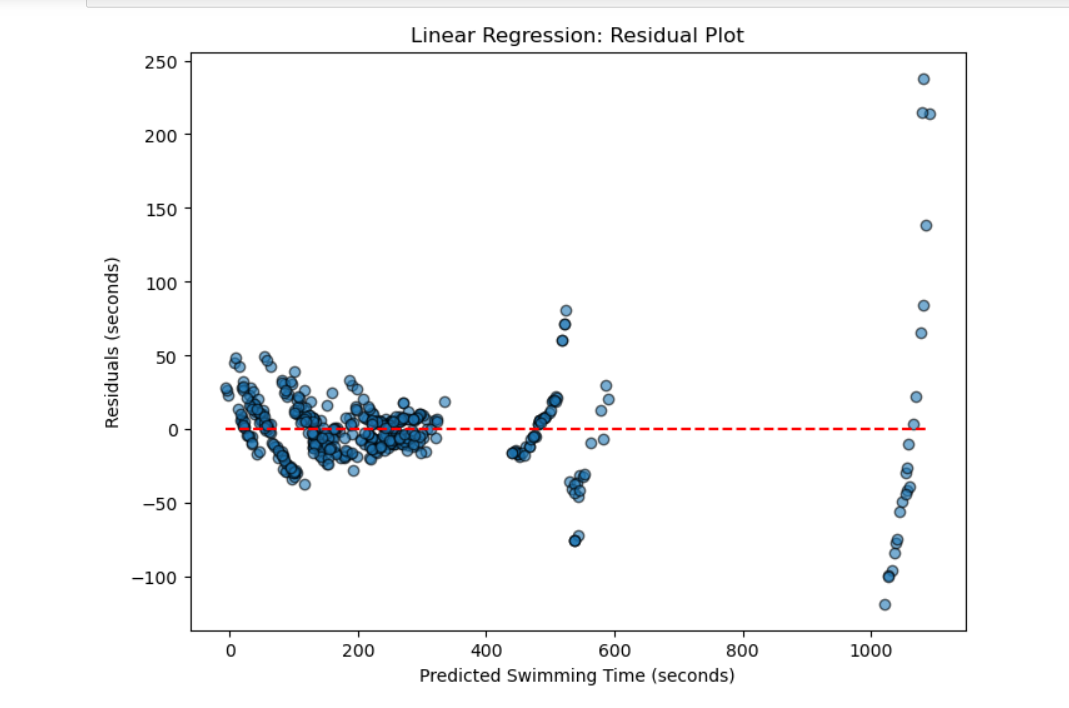
In this case, the residuals are roughly centered around 0, but there is some skew and spread, particularly toward the positive side, which indicates underestimation of swim times. This reflects that while the model was generally accurate for most cases, it occasionally produced significant over or under predictions, especially for longer swim times or less common events. These outliers could be due to the model’s inability to model non-linear relationships or handle event-specific variations.



**Figure - Actual vs. Predicted Swimming Times Using Linear Regression**

This scatter plot shows the relationship between the predicted swim times and the actual swim times produced by the linear regression model. The red dashed line represents the ideal scenario where predictions exactly match actual values, i.e., a perfect prediction. The closer the data points are to this line, the better the model is performing.

In this case, most points lie close to the line for swim times under 600 seconds, indicating good accuracy for shorter events. However, the spread increases for longer events, where several predictions deviate more from the line. This suggests that while the linear model performs fairly well for shorter or average-length events, it struggles to maintain accuracy as race duration increases. This outcome highlights the model's limitations in capturing non-linear patterns present in long-distance performance data.



**Figure - Residual Plot for Linear Regression Model**

The residual plot displays the difference between the actual and predicted swim times, i.e., residuals plotted against the predicted swim time. Ideally, in a good performing model, these residuals should be evenly scattered around zero with no obvious pattern, which would indicate consistent prediction error across all values.

In this plot, we can see that residuals are somewhat centered around the red zero line for mid range swim times 200–600 seconds, but there is noticeable spread and clustering at the extremes. In particular, larger residuals appear for long-distance predictions, suggesting that the linear model becomes less reliable for higher swim durations. This uneven distribution confirms that the model doesn’t fully capture the complexity of swimming performance across different event lengths

These visualisations helped in the better understanding of the model performance and limitations of the linear regression model. While it provided a reasonable fit for the majority of races, its residual patterns and deviations in longer events suggest that more advanced models, such as neural networks or tree-based methods, are better suited for capturing the full complexity of swimmer performance.

## 4.4 Multilayer Perceptron Neural Network

After establishing a baseline with Linear Regression, we moved to a more advanced model, the Multilayer Perceptron (MLP), which is a type of neural network. MLPs are well-suited to problems like this, where multiple variables may interact in nonlinear and complex ways. Unlike linear models, MLPs are capable of capturing curved or hidden patterns that are not immediately obvious.

The main goal was to improve prediction accuracy by using a model that could better understand relationships between features like event type, swimmer ranking, and race distance.

The Multilayer Perceptron (MLP) model was implemented using the MLPRegressor class from the sklearn.neural\_network module. The entire pipeline was constructed using Pipeline and ColumnTransformer from scikit-learn to manage preprocessing and model fitting in a single, unified process.

* Numerical features (Ranking\_numeric, Distance) were standardized using StandardScaler().
* Categorical features (Sex, Event) were encoded using OneHotEncoder(handle\_unknown='ignore').
* The model was trained on a dataset with outliers removed to improve learning consistency and accuracy.
* The MLP model was optimized with the following hyperparameters:

hidden\_layer\_sizes=(8, 4)

learning\_rate\_init=0.001

alpha=0.0001 (L2 regularization)

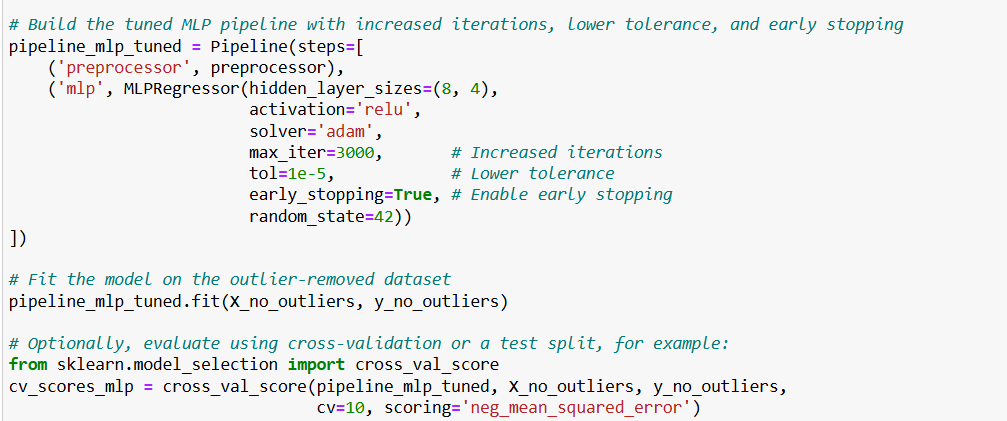
max\_iter=2000 with early\_stopping=True

tol=1e-5 for improved convergence precision

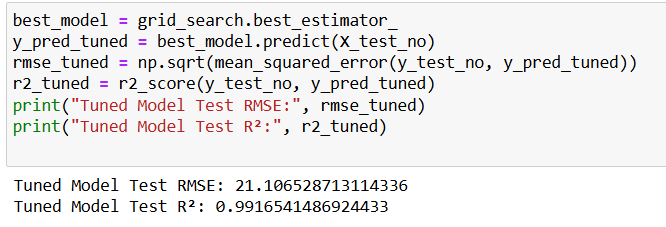
* Evaluation was done using:
  + - Root Mean Squared Error (RMSE) via mean\_squared\_error()
    - R² Score via r2\_score()
    - 10-fold cross-validation using cross\_val\_score()

These settings were determined after experimenting with different architectures and using GridSearchCV to find the best-performing combination of hyperparameters.

Below is the Python code used to define the tuned MLP pipeline, fit it to the outlier-removed dataset, and evaluate it using both test data and cross-validation



To calculate test set performance using the best tuned model from GridSearchCV



This code illustrates the steps taken to implement and evaluate the MLP model. It reflects both the improved architecture via hyperparameter tuning and the model’s strong performance, especially in comparison to the Linear Regression baseline.

Our initial attempt with a simple MLP architecture, one hidden layer with 4 neurons did not perform well. The model generated convergence warnings and had a high RMSE, suggesting it struggled to learn from the data.To improve the performance, we made several changes

We increased the network's capacity by using two hidden layers with sizes (8, 4), meaning the first layer had 8 neurons and the second had 4. This gave the model more room to learn deeper patterns in the data.

Training Optimization:

Increased max\_iter from 500 to 2000 to allow more training cycles

Reduced tol (tolerance) to 1e-5, which made the model more precise in deciding when to stop training

Enabled early stopping, which automatically halts training if the model's performance stops improving and helped in preventing overfitting

The hyperparameter tuning was also used by implementing GridSearchCV to test different settings and find the best-performing configuration. The optimal parameters were

hidden\_layer\_sizes: (8, 4)

learning\_rate\_init: 0.001

alpha: 0.0001

max\_iter: 2000

These hyperparameters were selected because they directly influence how the model learns from data. The hidden\_layer\_sizes control the structure of the neural network in this case, two layers with 8 and 4 neurons allowed the model to learn more complex relationships between inputs and outputs. The learning\_rate\_init determines how quickly the model updates its weights during training; a smaller value (0.001) provides more stable learning. The alpha parameter is used for regularization, which helps prevent overfitting by discouraging the model from becoming too complex. At the end, max\_iter was increased to 2000 to give the model more iterations to learn effectively, especially after the complexity of a deeper architecture.

The final model was trained on the same preprocessed dataset described earlier, using the Time\_seconds column as the target variable.

After training and testing the MLP model, the following results were observed:

Test RMSE (~21.11 seconds):

This shows that the model’s predictions were off by about 21 seconds on average a significant improvement over the Linear Regression model, which had an RMSE of 40 seconds.

For longer races (like 400m, 800m, or 1500m), this level of error is relatively acceptable, considering those races often take 300–1000 seconds to complete. However, for shorter races, an error of 21 seconds is still quite large, meaning the model still struggled with precision on sprint events.

R² Score (0.99165):

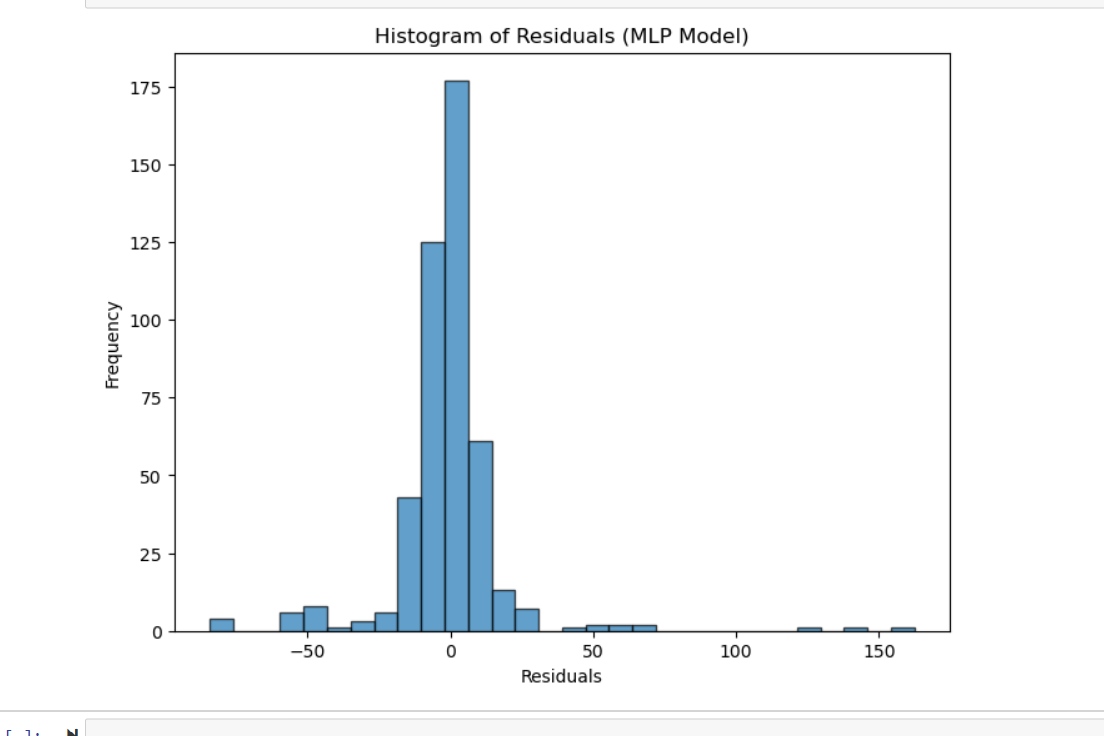
This value is very close to 1, indicating that the model was able to explain nearly all the variation in swim times based on the selected input features. A high R² means the model successfully understood how different variables, such as distance, swimmer ranking, and event type, contributed to performance. This level of fit suggests that the MLP was capable of capturing the hidden relationships in the data that the linear model missed.

Cross-Validation RMSE (~61.00 ± 54.49):

The model’s performance varied widely across the different data splits used in cross-validation. This variability suggests that the MLP was more sensitive to how the data was split and may be unstable. It likely performed better when the training set included more examples of specific events or race distances.

The MLP model more than doubled the accuracy compared to Linear Regression, reducing RMSE by almost 50%. It also demonstrated an ability to model complex, nonlinear patterns, which are common in swimming data, for instance, the way ranking influences swim time differently in a 50m sprint compared to a 1500m endurance event.

Some data visualisations were also created to have overlook how well the model is performing

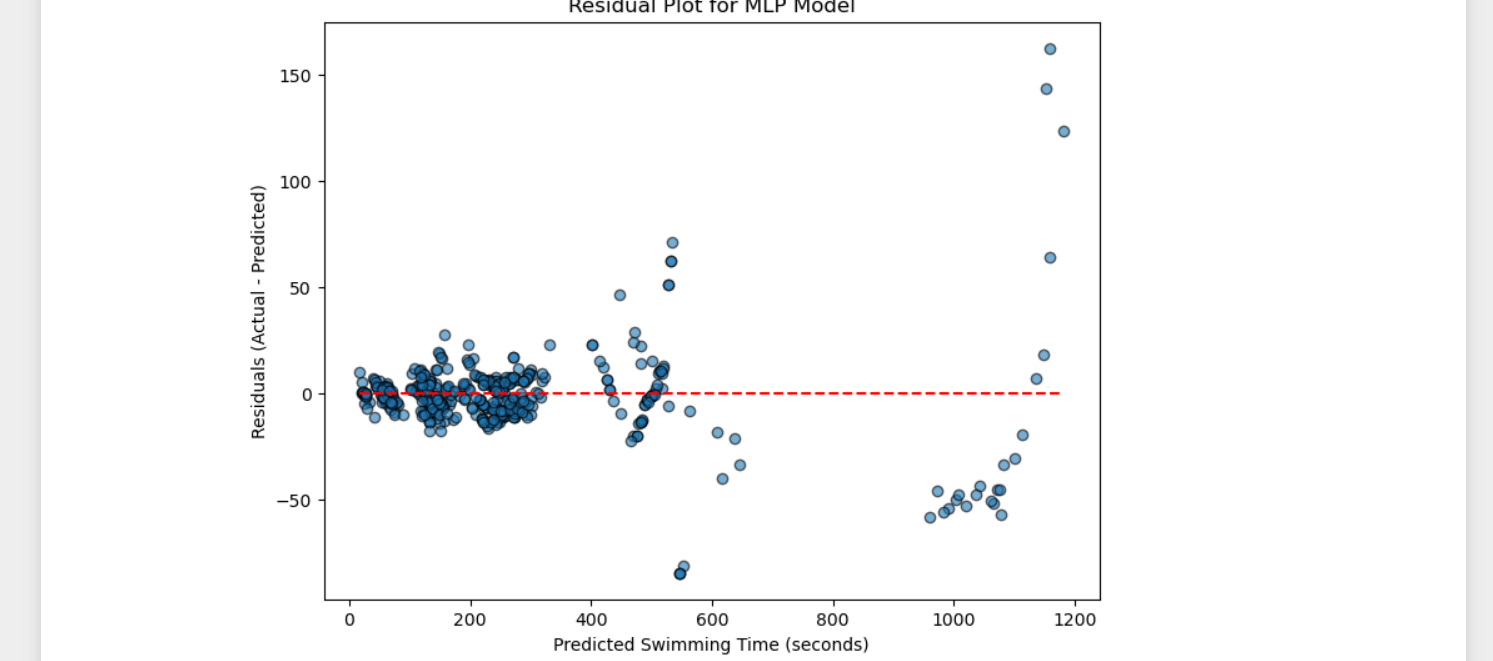


**Figure - Histogram of Residuals of MLP Model**

This histogram shows the distribution of residuals from the MLP model. A strong model will have a histogram that is centered around zero and shaped roughly like a bell curve, indicating that most predictions were close to the actual values and no major bias exists.

The residuals are centered at 0, with a high frequency of residuals falling between -20 and +20 seconds. While a few outliers extend further up to ±150 seconds, these are rare. The overall shape of the distribution confirms that most of the model’s predictions were very accurate, and the spread of error was smaller than that of the Linear Regression model.

This histogram further reinforces the strength of the MLP model, showing that the majority of predictions were tightly clustered around the correct values.

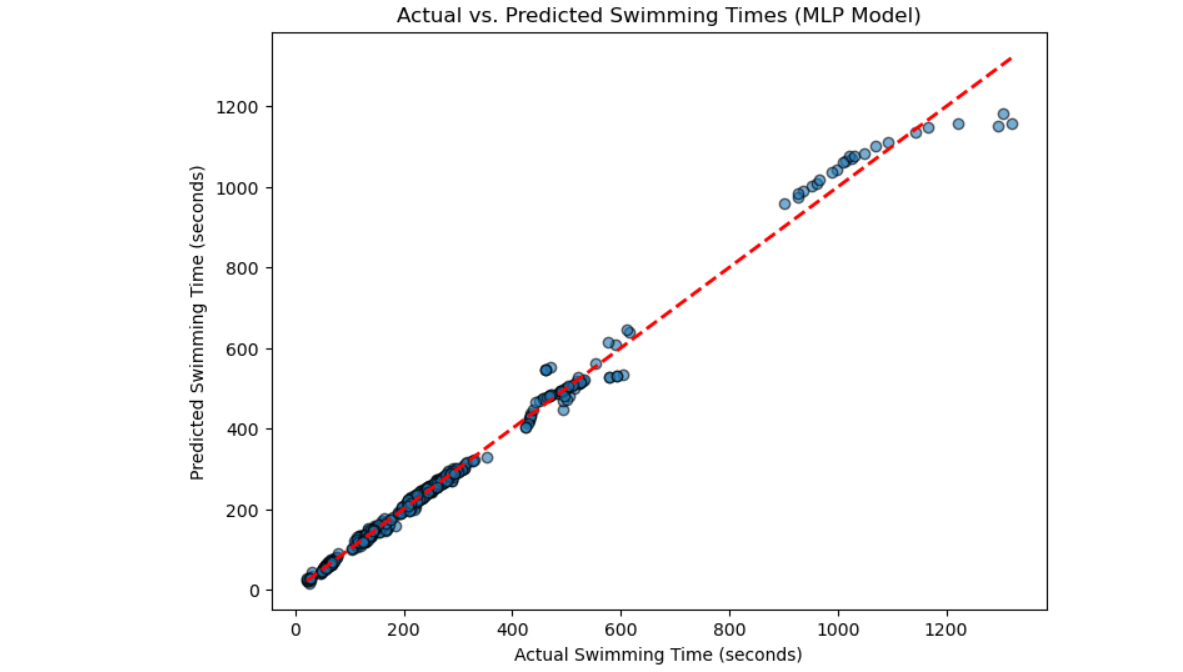


**Figure - Residual Plot for MLP Model Predictions**

The residual plot displays the difference between actual and predicted swim times plotted against the predicted swim time. In an ideal model, residuals should be evenly spread around the zero line, without a clear pattern or clustering, indicating stable and unbiased performance.

The residuals in this plot are mostly concentrated around zero, particularly for races under 600 seconds. However, some wider spread and slight vertical clustering appear as the predicted times increase, especially in longer events. This shows that while the model was generally accurate, prediction errors increased slightly for longer-distance races, a challenge common to many models due to reduced data availability and greater time variability in endurance events.

Overall, this plot supports the conclusion that the MLP model produced low and stable error values, especially in the race ranges where most data was concentrated.



**Figure - Actual vs. Predicted Swimming Times Using MLP Model**

This scatter plot compares the actual swim times with the predicted swim times genehidden\_layer\_sizesrated by the MLP model. The red dashed line represents the ideal line where predictions would exactly match the actual values.

In this case, the points closely follow the line, especially in the 0–600 second range, showing that the MLP model achieved strong accuracy across a wide variety of races. While a few deviations appear for longer swim times, particularly above 1000 seconds, overall the spread is consistent, a clear improvement compared to the Linear Regression model.

This visualisation confirms that the MLP model was capable of capturing the complex patterns in the data, producing predictions that align very closely with real swim performances.

However, while the performance on the test set was excellent, the high variation during cross-validation suggests that this model may struggle with generalization, especially when trained on imbalanced or sparse subsets of the data. It’s a sign that although MLPs are powerful, they are also more dependent on data quality and size.

Overall, the MLP model proved to be a significant step forward in predicting swimming performance. It showed that machine learning can successfully uncover complex patterns in athlete data, but it also highlighted the importance of careful model tuning and evaluation.

## 4.5 XGBoost Regression

The final model to predict the accuracy of time was XGBoost. To develop the XGBoost regression model, we used the XGBRegressor class from the xgboost library. A custom pipeline was built using Pipeline and ColumnTransformer from scikit-learn, allowing us to preprocess data and fit the model in one streamlined process.

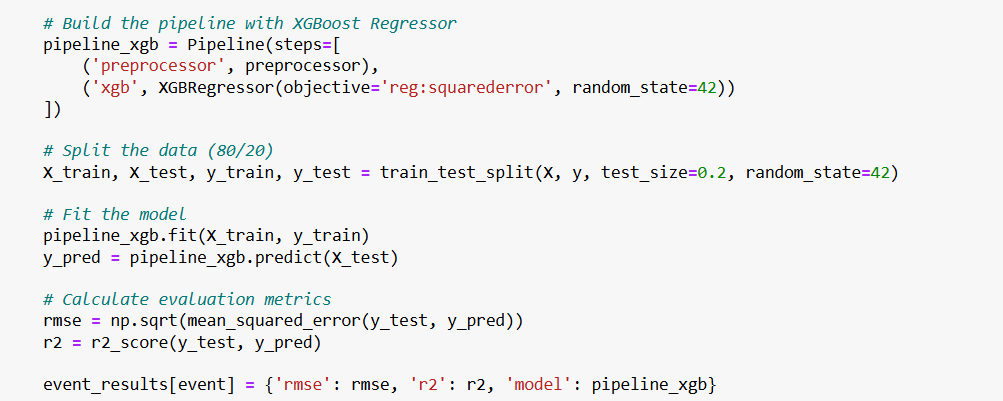
* Numerical features (Ranking\_numeric, Distance) were scaled using StandardScaler().
* Categorical variables (Sex) were encoded using OneHotEncoder().
* Data was split using train\_test\_split() (80% training, 20% testing).
* We evaluated model performance using

Root Mean Squared Error (RMSE) via mean\_squared\_error()

R² Score via r2\_score(), which measures the proportion of variance explained

Each swimming event with at least 50 samples was modeled separately to enable event-specific learning, resulting in custom XGBoost models for each event type.

The pipeline below ensured consistent preprocessing and allowed us to train and evaluate event-specific models with minimal duplication in code structure.



### 4.5.1 Per-Event Performance

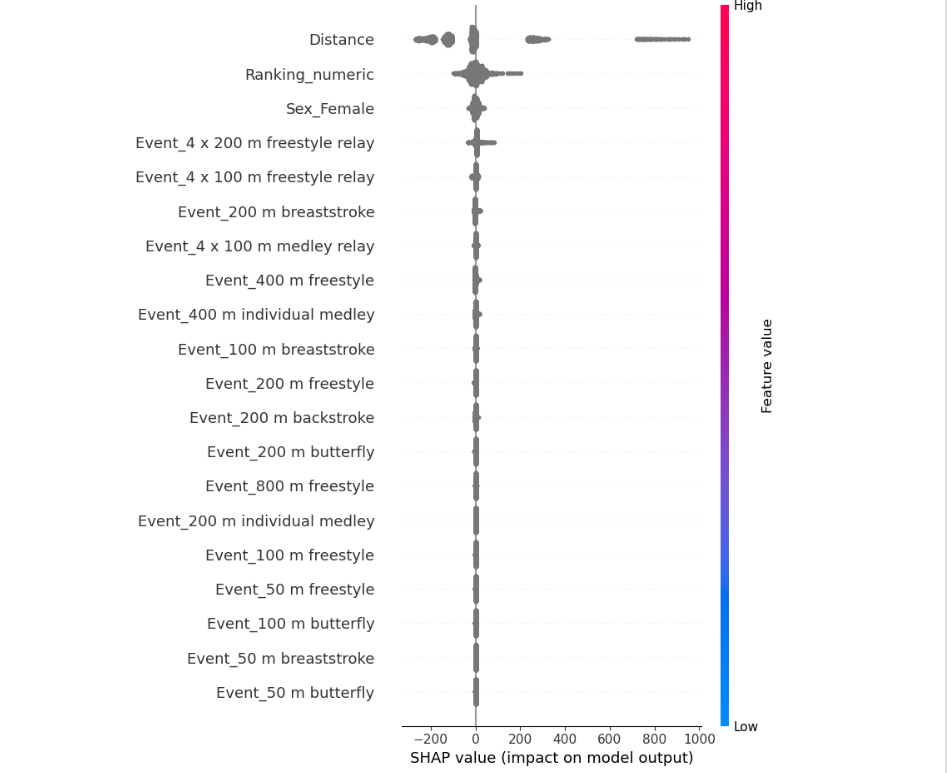
The table below summarizes XGBoost’s predictive performance for each event using RMSE and R²:

| **Event** | **Test RMSE (s)** | **R² Score** |
| --- | --- | --- |
| 4 x 100 m medley relay | 0.01 | 1.0000 |
| 4 x 100 m freestyle relay | 0.91 | 0.9983 |
| 4 x 200 m freestyle relay | 0.00 | 1.0000 |
| 200 m breaststroke | 2.00 | 0.9885 |
| 400 m freestyle | 3.62 | 0.9876 |
| 200 m backstroke | 6.74 | 0.8607 |
| 800 m freestyle | 11.72 | 0.9745 |
| 100 m freestyle | 0.55 | 0.9918 |
| 200 m freestyle | 2.82 | 0.9533 |
| 100 m breaststroke | 0.33 | 0.9956 |
| 200 m individual medley | 1.28 | 0.9826 |
| 1500 m freestyle | 17.40 | 0.9802 |
| 200 m butterfly | 1.16 | 0.9885 |
| 400 m individual medley | 2.14 | 0.9936 |
| 100 m butterfly | 0.51 | 0.9897 |
| 100 m backstroke | 0.60 | 0.9702 |
| 50 m freestyle | 0.20 | 0.9830 |

These results show exceptionally strong accuracy across most events. RMSE values were often below 1 second for sprint events and remained acceptably low for endurance races. The consistently high R² scores indicate that the model explained nearly all of the variation in swim times.

To better understand how the XGBoost model arrived at its predictions, we used SHAP (SHapley Additive Explanations) a method that quantifies the contribution of each feature to individual predictions. The SHAP summary plot revealed that Distance was the most influential feature in predicting swim time, which aligns with expectations as longer races naturally result in higher completion times. The feature Ranking\_numeric also played a significant role, suggesting that better-ranked swimmers tend to achieve faster times across events. Other features such as Sex and specific Event types contributed modestly but meaningfully, indicating that the model accounted for gender-based differences and the technical nature of different races.

Overall, the SHAP analysis confirmed that the model prioritized variables in a way that reflects real-world swimming dynamics, making it both accurate and interpretable.

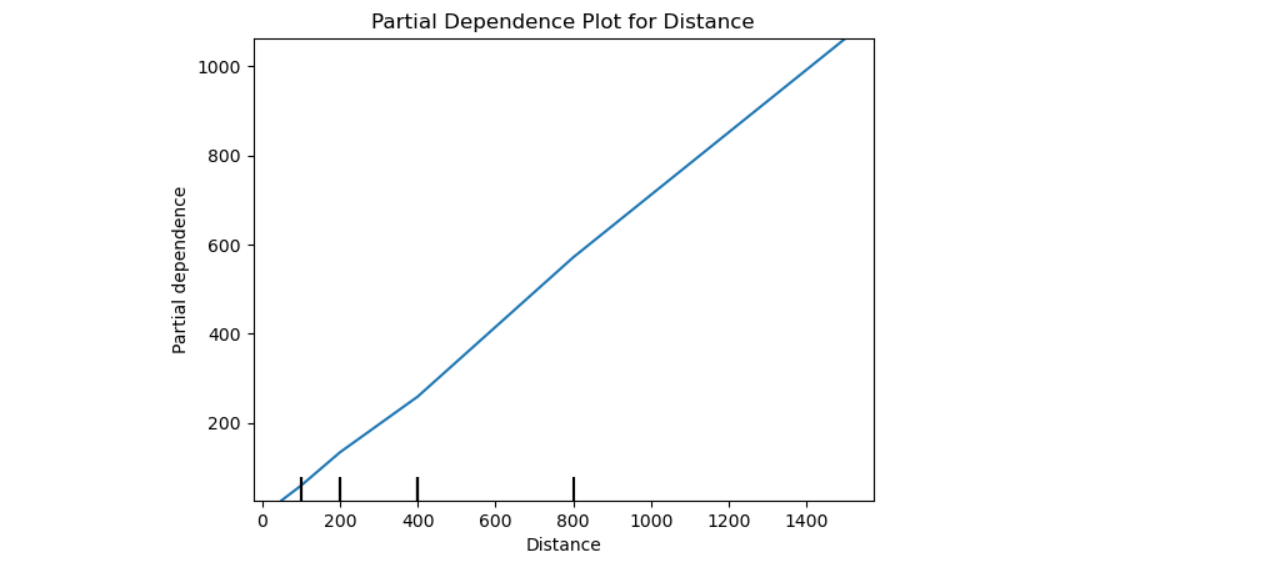


**Figure - SHAP Summary Plot for XGBoost Model**

This plot visualizes the contribution of each feature to the model’s predictions. Each dot represents a sample, with horizontal placement showing the direction and magnitude of impact on predicted swim time. Color represents the feature value (blue = low, red = high).

To further validate the model’s understanding of race structure, a Partial Dependence Plotwas generated for the Distance feature. This plot shows how predicted swim time changes as race length increases, while holding other features constant. The graph displayed a clear linear increase in predicted swim time as the event distance increased.

This confirms that the model correctly captured the basic relationship between race distance and swim duration — a key component of real-world performance. The smooth curve and lack of irregularities in the PDP also suggest that the model did not overfit or apply erratic logic.



**Figure – Partial Dependence Plot for Distance**

This graph illustrates the isolated effect of the Distance feature on predicted swim time. The upward trend confirms that longer races were predicted to take longer, consistent with expected performance behavior.

**Key Finding**

The XGBoost regression model demonstrated superior performance compared to both Linear Regression and MLP. Its ability to capture nonlinear interactions, model per-event nuances, and explain predictions through SHAP made it the most accurate and interpretable model in the study.

* High accuracy across all events, including sprint and endurance races
* Excellent generalization with minimal error and high R² values
* Transparent model logic, made possible through SHAP and PDP

This makes XGBoost highly suitable for practical applications such as performance prediction, race simulation, and athlete development analysis.

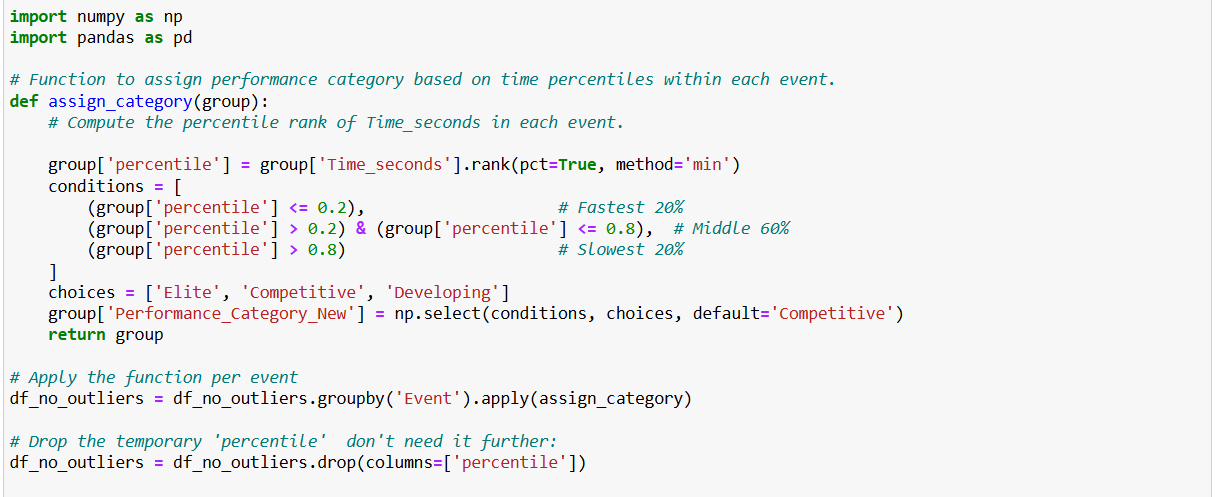
## 4.6 Random Forest Classifier

After evaluating models for swim time prediction, the next goal was to classify swimmers into performance tiers based on their race characteristics and results. This involved reframing the problem into a classification task, where each swimmer was labeled as Elite, Competitive, or Developing based on their relative performance within each event.

This classification approach allowed us to group swimmers not just by absolute time or rank, but by their percentile performance within event categories, making the model more realistic and useful for talent identification or training prioritization.

The classification model was built using the RandomForestClassifier from the sklearn.ensemble module. A full pipeline was constructed with Pipeline and ColumnTransformer to handle preprocessing and model training.

Target Variable: Performance\_Category\_New, derived by ranking swimmers within each event based on swim time and assigning:



Input Features used for prediction:

Ranking\_numeric: Swimmer’s final rank (numeric)

Distance: Event length in meters (numeric)

Sex: Gender of the swimmer (categorical)

Event: Type of event (categorical)

Preprocessing steps:

StandardScaler() for numerical features

OneHotEncoder(handle\_unknown='ignore') for categorical features

The dataset used for training was filtered to exclude outliers, and an 80/20 split was applied using train\_test\_split() for training and testing.

**Model Tuning and Evaluation**

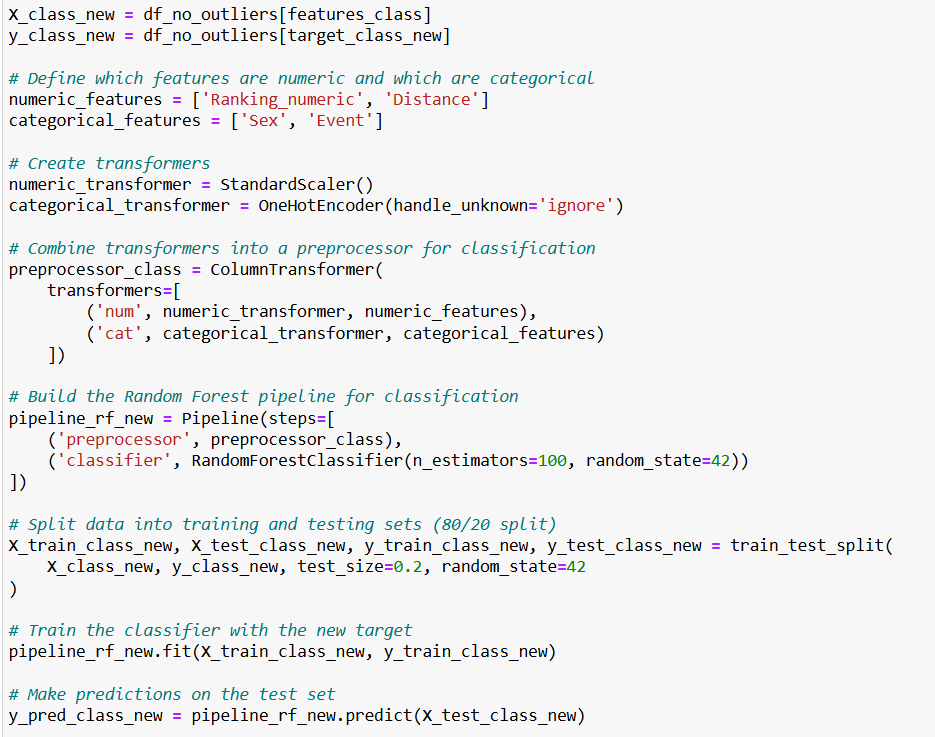
To ensure optimal performance, the model was tuned using GridSearchCV with a 5-fold cross-validation strategy. The hyperparameter grid included:

* n\_estimators: [100, 200, 300]
* max\_depth: [None, 10, 20, 30]
* min\_samples\_split: [2, 5, 10]
* min\_samples\_leaf: [1, 2, 4]
* class\_weight: [None, 'balanced']

These hyperparameters were selected because they directly control how the model builds and balances the decision trees. For example, increasing max\_depth allows the model to learn deeper patterns, while using class\_weight='balanced' adjusts for any imbalances in category frequencies.

The best-performing model from the grid search was selected for final testing and evaluation.

**Random Forest Pipeline and Tuning**

****

This code sets up the complete model pipeline, performs hyperparameter tuning, and evaluates classification performance.

**Model Results**

After training and evaluation, the model achieved the following results:

* Test Accuracy: 91.4%

This means that the model correctly predicted the performance category for around 9 out of 10 swimmers in the test set.

* Cross-Validation Accuracy: Varied across folds but consistently high, 88–93%
* Classification Report:

The precision, recall, and F1-scores were strong across all three performance categories. The model showed slightly higher accuracy in predicting Elite and Competitive swimmers compared to Developing, which likely had fewer examples in the data.

* Confusion Matrix:

Most predictions were concentrated along the diagonal, confirming that the model made accurate classifications. A small number of swimmers labeled "Competitive" were misclassified as "Developing", which is common when class boundaries are close.

### 4.6.1 Real-World Use of Random Forest Model

This Random Forest model effectively categorized swimmers by performance tier based on event features and swimmer statistics. Unlike regression, which focused on predicting specific times, this approach allowed us to segment athletes into interpretable groups — a technique useful for coaches, scouts, and performance analysts.

The classification could help:

* Identify top-performing (Elite) athletes per event
* Provide tailored training plans for Developing swimmers
* Predict how a swimmer might be expected to perform in unfamiliar events

The model also supports generating useful outputs such as:

* Top 3 swimmers per category
* Predicted category distributions
* Event-wise performance summaries

The Random Forest Classifier proved to be a highly accurate and interpretable tool for grouping swimmers into performance tiers. By using both rank-based and event-based information, it created a flexible model that generalizes well to various race formats.

This classification task demonstrated how machine learning can go beyond time prediction, offering practical insights for talent development, training management, and performance tracking in competitive swimming.

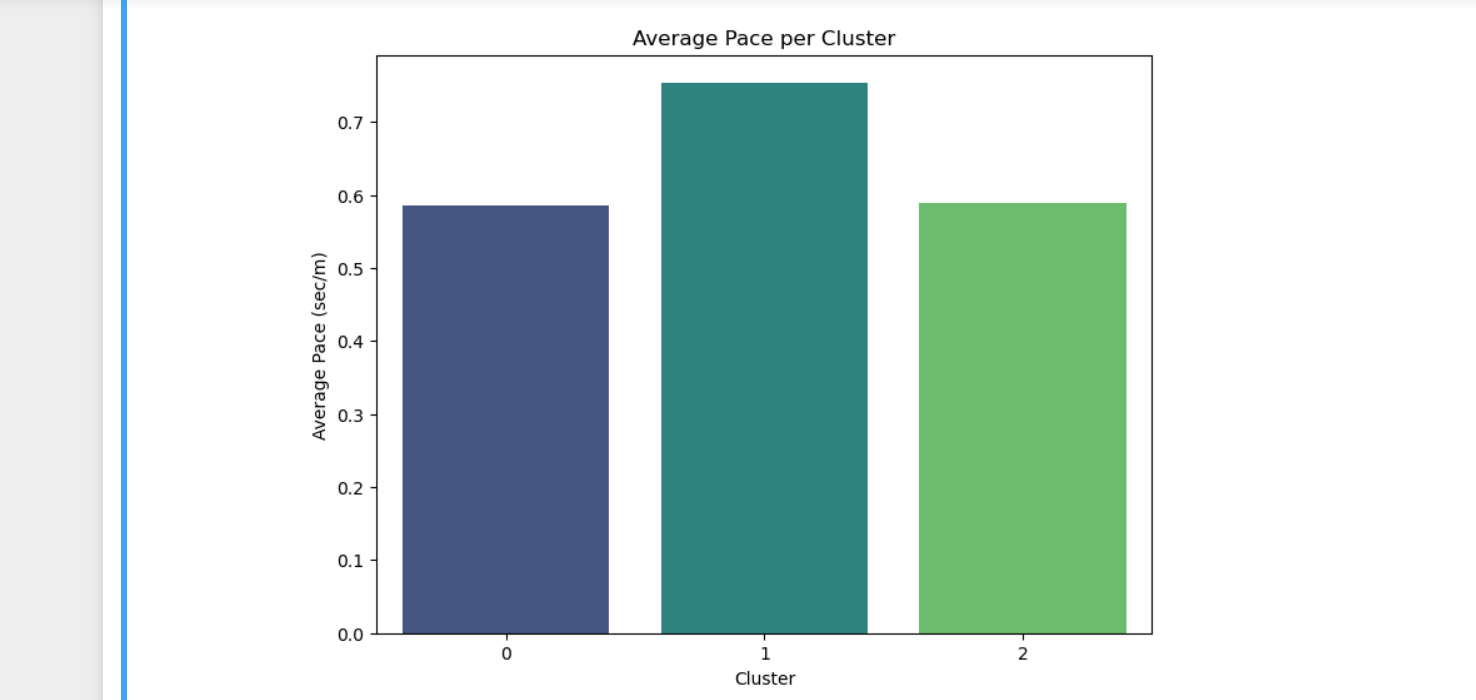
## 4.7 Clustering Model

After building predictive models, we introduced an unsupervised learning approach to explore hidden patterns within the swimming performance data. The goal was to group swimmers into natural performance categories without using predefined labels such as rankings. This allowed us to objectively analyze whether swimmers could be grouped based on similar race characteristics (e.g., pace, distance, or time).

We chose KMeans clustering from the sklearn.cluster module due to its simplicity, speed, and suitability for numerical data.

The clustering process was structured as follows:

* Feature Selection: We used key numerical variables:
  + - Distance – length of the race
    - Time\_seconds – swim time in seconds
    - Pace – calculated as Time\_seconds / Distance (seconds per meter)
* Scaling: Since K-Means is sensitive to feature magnitude, the features were standardized using StandardScaler().
* Clustering: KMeans clustering was applied with n\_clusters=3 to classify swimmers into three performance groups.
* Cluster Labels: These clusters were then interpreted based on swim time and pace metrics:
  + - Cluster 0 (Elite): Shortest swim times, best paces, lower variability. Likely includes high-performing or competitive swimmers.
    - Cluster 1 (Competitive): Moderate pace and performance with some inconsistency. Possibly includes club-level or developing swimmers.
    - Cluster 2 (Developing): Longer swim times with consistent but slower paces. May consist of new or recreational athletes

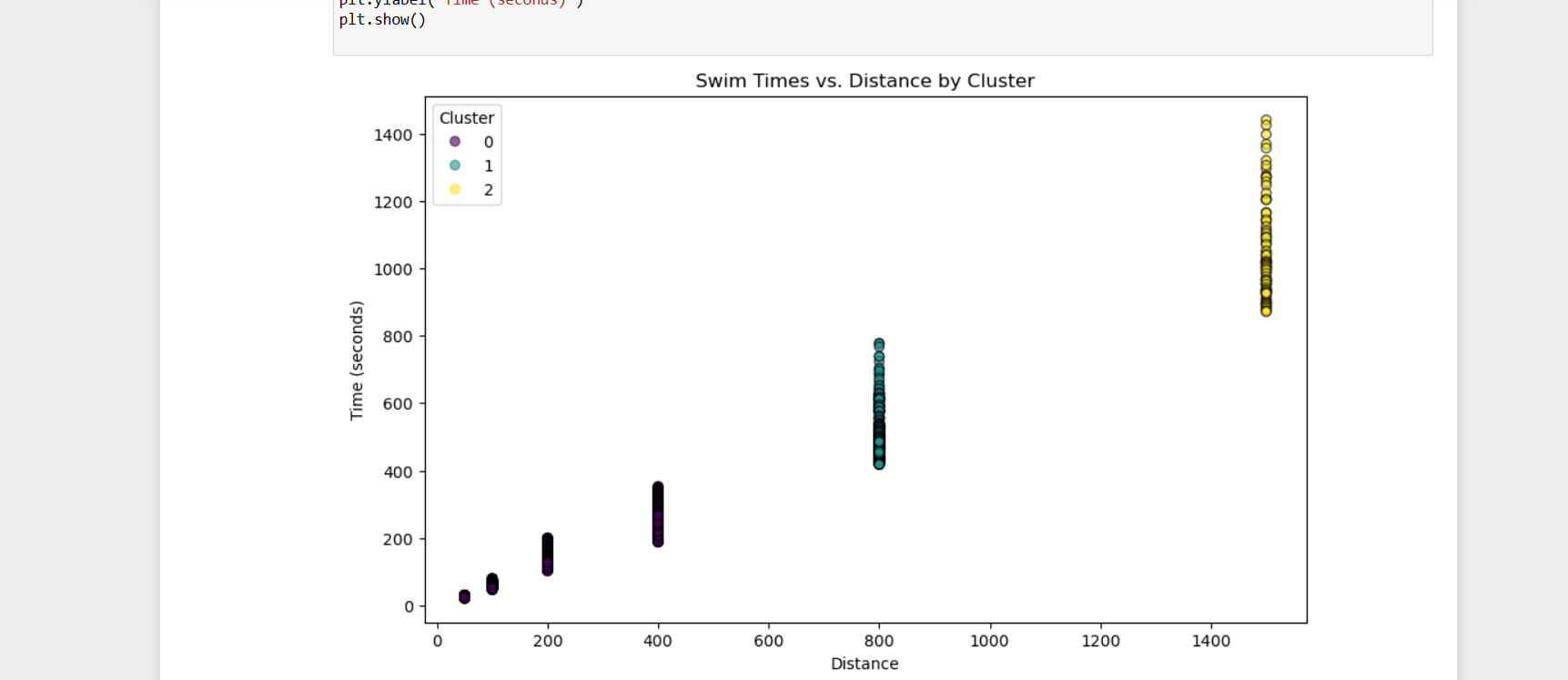


**Figure - Bar Chart of Average Pace by Cluster**

**Bar Plot: Average Pace per Cluster**

This bar chart reinforces the previous plot, showing:

* Cluster 1 has the slowest average pace.
* Clusters 0 and 2 exhibit better efficiency, with Cluster 0 being the fastest overall.

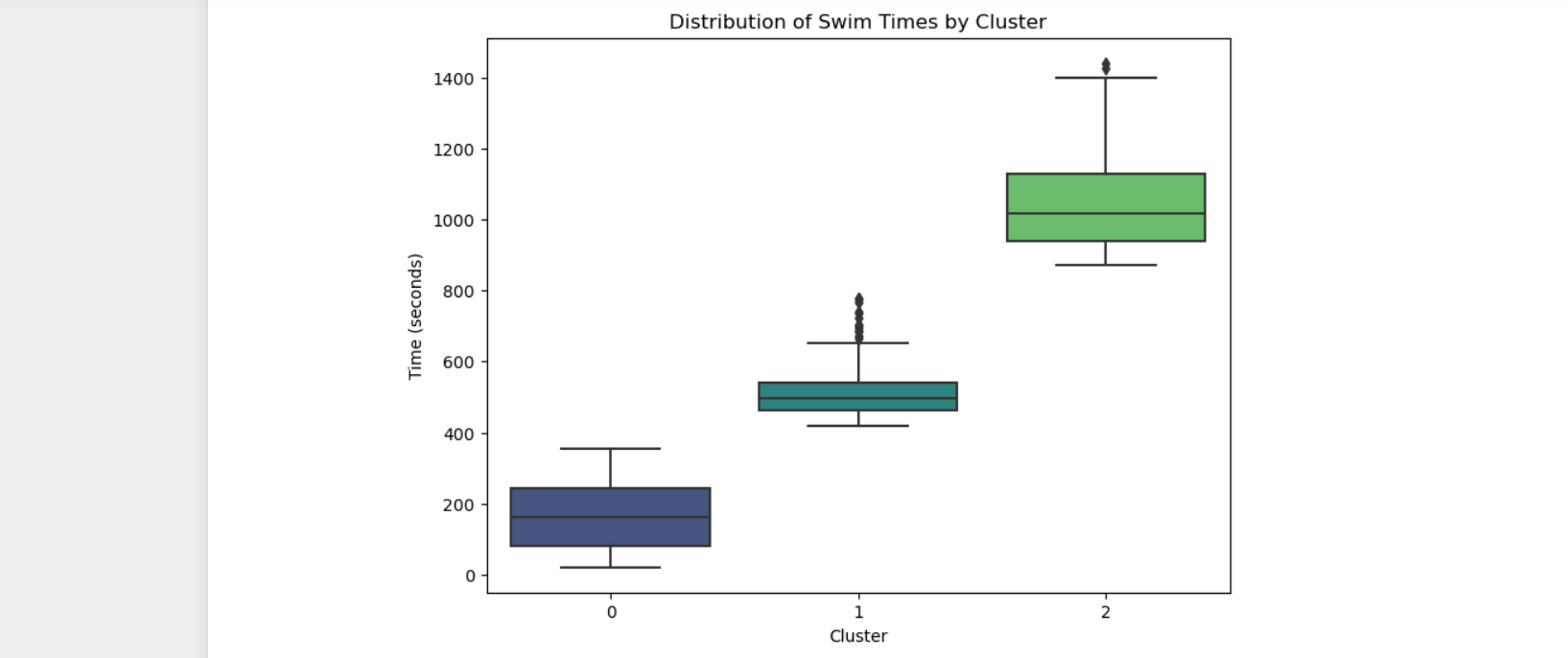


**Figure - Swim Time vs. Distance by Cluster**

**Scatter Plot: Swim Time vs Distance by Cluster**

This scatter plot visually demonstrates how swim times differ across distances and clusters. Each color represents a different cluster:

* Cluster 0 (dark purple): Generally faster swim times at all distances
* Cluster 1 (blue/teal): Moderate times
* Cluster 2 (yellow): Slower performances, especially at longer distances

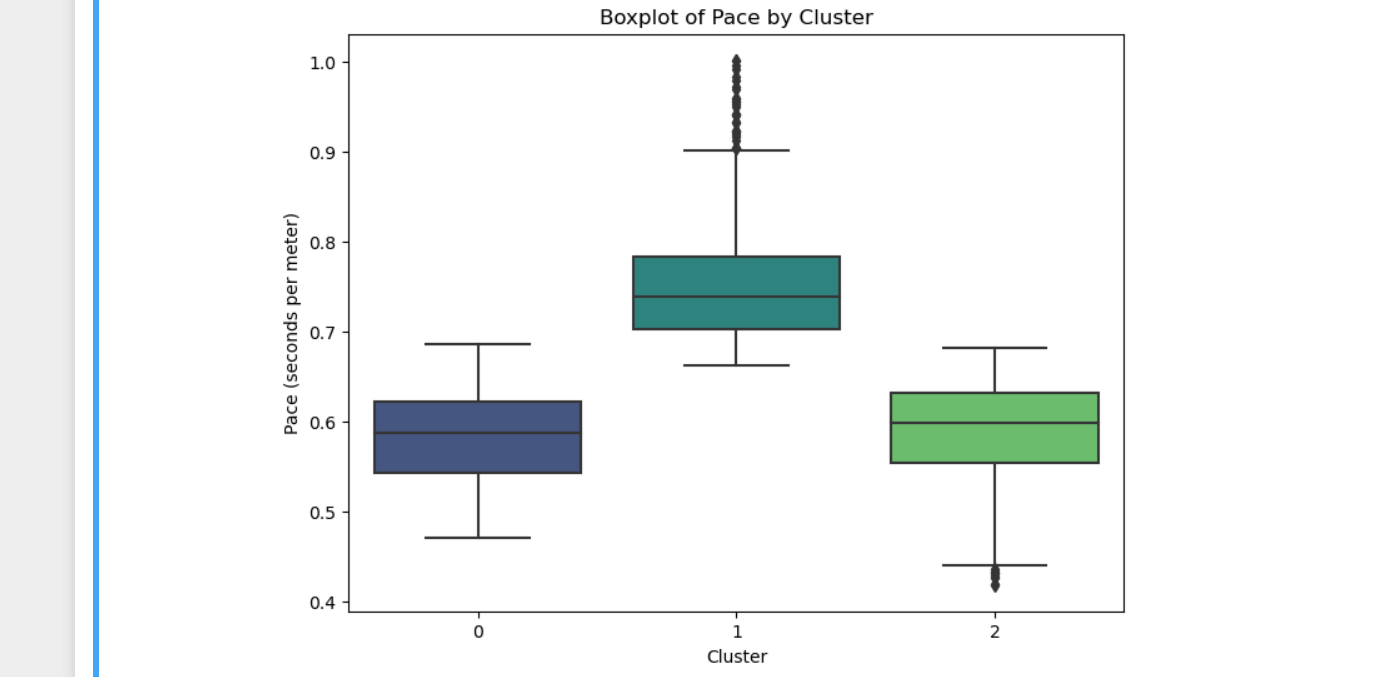


**Figure - Boxplot of Swim Time Distribution by Cluster**

**Boxplot: Swim Time by Cluster**

This boxplot shows the spread and median swim times within each cluster. The visualization reveals a clear upward trend in median swim times from Cluster 0 to Cluster 2:

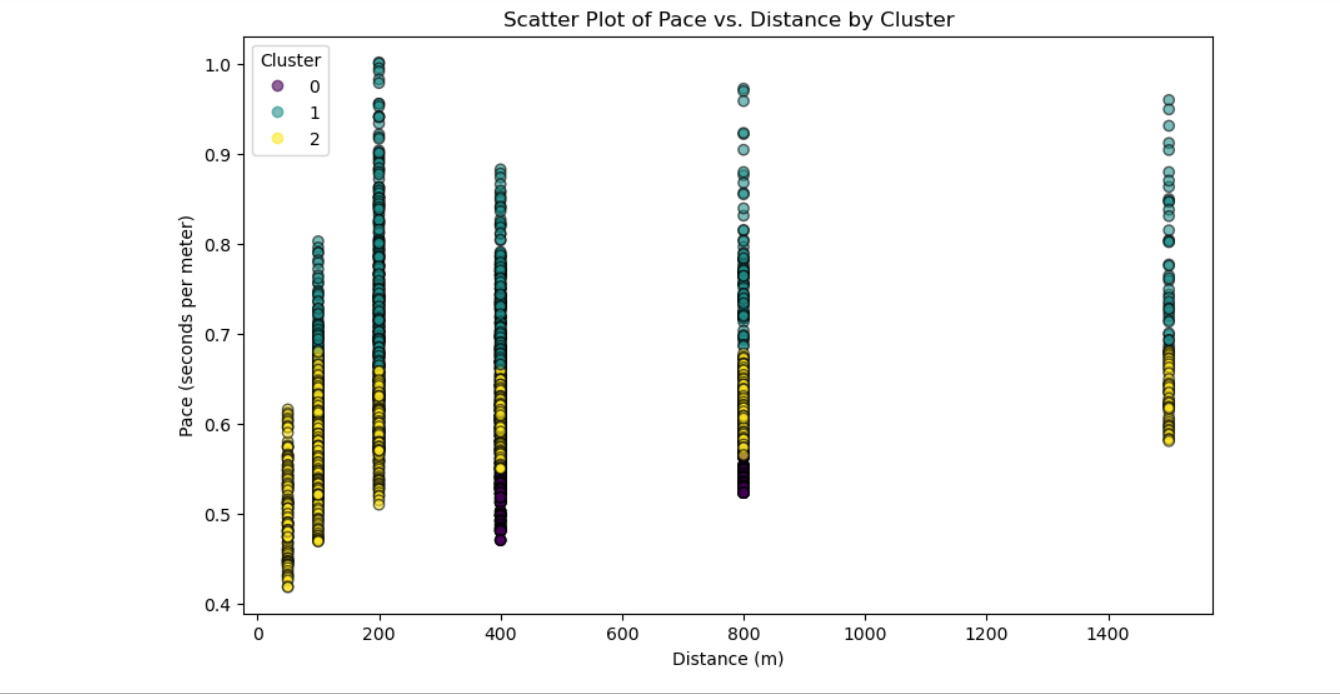
* Cluster 0: Shortest times
* Cluster 1: Mid-range
* Cluster 2: Longest durations



**Figure - Boxplot of Swimming Pace by Cluster**

Here, we evaluate how efficient swimmers are (in seconds per meter).

* Cluster 0 and Cluster 2 have similar median pace values, with Cluster 2 showing tighter consistency.
* Cluster 1 stands out with slower average pace and higher variance indicating it contains swimmers with inconsistent performance.



**Figure - Clustering of Swimming Pace vs. Distance**

**Scatter Plot: Pace vs Distance by Cluster**

This plot provides another perspective on how pace varies across event distances and performance categories. It reveals that:

* Pace generally improves with longer distances in the better-performing clusters.
* Some overlap exists, but the trend supports the cluster distinctions.

In the visualisations, a few data points fall far from the cluster centers, particularly in longer-distance races. These outliers represent atypical performances, such as exceptionally fast or slow times, which are not uncommon in real-world data. Their presence did not significantly impact the interpretability or formation of core clusters, but is worth noting as part of the dataset’s natural variability.

### 4.7.1 Notable Swimmers Identified in High-Performance Groups

To add interpretability and validate the clustering and classification results, we analyzed the names of swimmers within the fastest-performing groups (clusters and predicted categories). This qualitative inspection revealed that many world-renowned athletes were correctly grouped into high-performance segments.

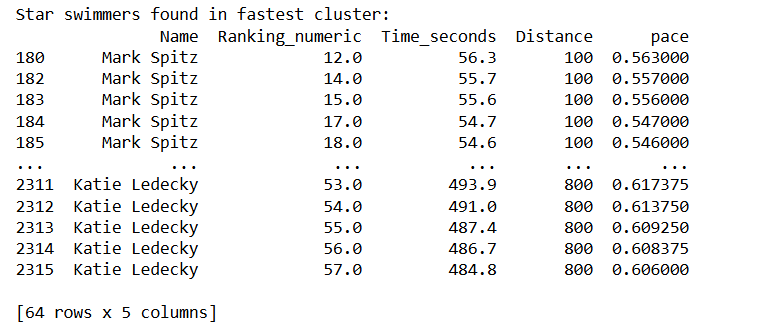
#### **4.7.1.1 Fastest Clusters Included:**

* Mark Spitz dominated the 100m events with consistent sub-57-second finishes.
* Katie Ledecky appeared multiple times in long-distance events such as 800m freestyle, all with exceptional times under 495 seconds.
* Michael Phelps, Ryan Lochte, Ian Thorpe – Clustered around the 400–800m freestyle distances, showcasing elite-level performance.
* Anthony Robinson and Lenny Krayzelburg featured prominently in 50–100m sprint events, reinforcing their specialist sprint capabilities.
* Penelope Heyns and Rebecca Soni identified mid-distance breaststroke-focused clusters with sub-150-second times.

#### **4.7.1.2 Elite Classification Category (Based on Pace and Ranking):**

* David Berkoff, Igor Polyansky, Geoff Huegill, and Zoe Baker – All achieved elite designation based on pace-adjusted clustering and classification.
* These athletes consistently demonstrated low time-per-meter pace and high rankings, aligning with their competitive success in real-world performance.

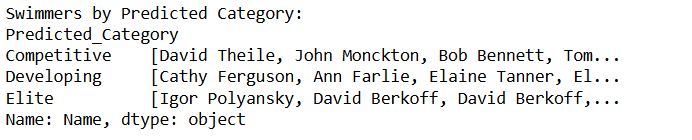
This alignment of elite athletes within top-performing clusters and categories serves as strong validation of the models’ ability to meaningfully segment swimming performance. It also enhances model transparency, giving insight into which individuals represent excellence within each analytical grouping.



The above output from the model presents a subset of star athletes, such as Mark Spitz and Katie Ledecky, who were consistently assigned to the fastest cluster based on time and pace. Their inclusion provides external validation of the model, as these individuals are globally recognized for their elite performances.

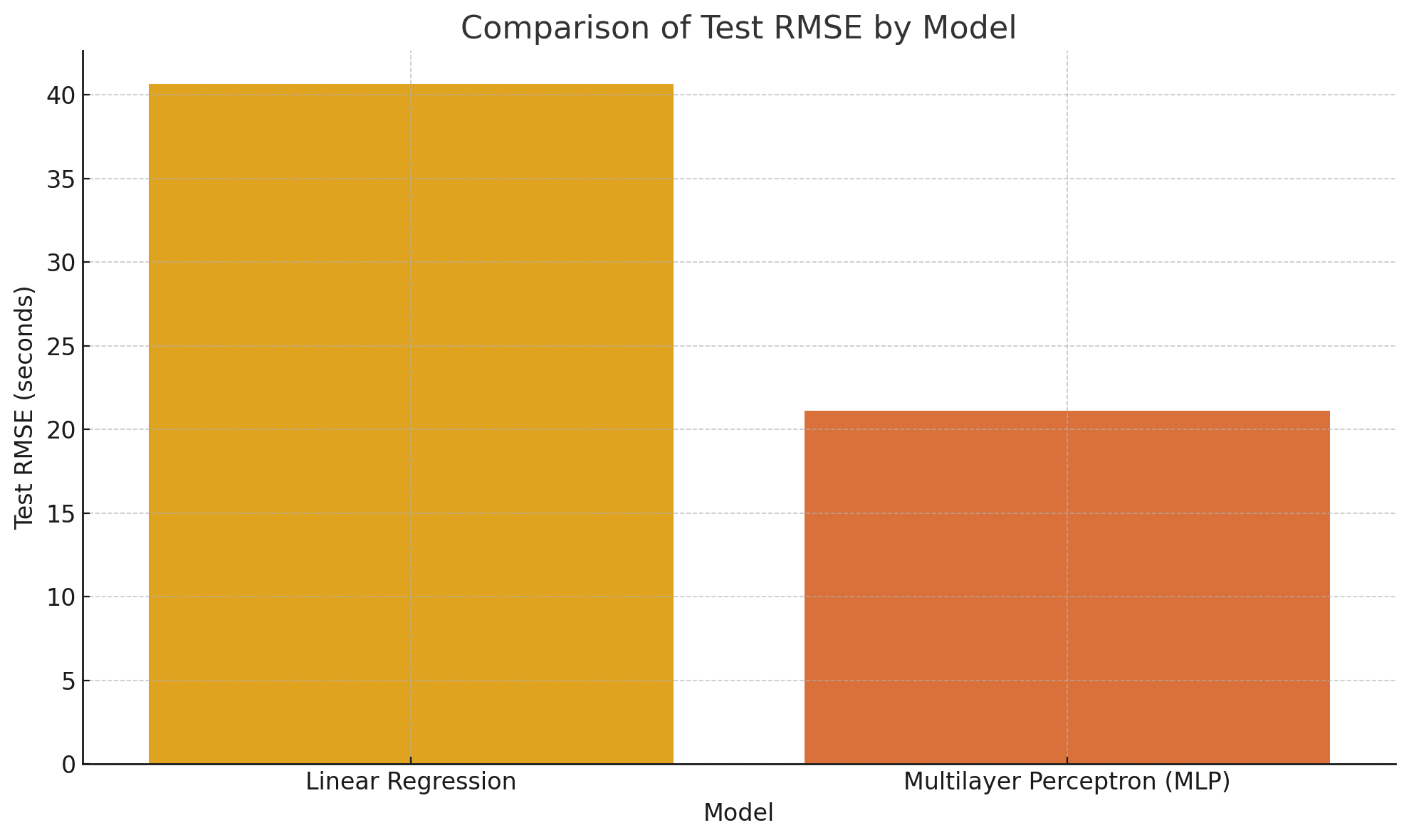


The output shows that the sorted members of the fastest cluster Cluster 2 by their competitive Ranking\_numeric, revealing top-tier athletes like David Theile, Inge de Bruijn, and Geoff Huegill. This reinforced that the cluster model successfully grouped swimmers who had both top rankings and low race times.



This output highlights the swimmers grouped by the predicted performance categories from the classification model. The Elite category notably includes names like David Berkoff and Igor Polyansky, again aligning model output with known international standards.

It demonstrates that the unsupervised and supervised models not only detected underlying performance trends but also aligned well with real-world athlete status, making them useful tools for talent identification, training tier classification, or benchmarking purposes. The clustering model provided strong, interpretable performance groupings that were further validated by the presence of legendary athletes in the highest-performing segments. These insights complement the classification model and show that swim performance naturally separates into competitive tiers, even without labeled supervision.

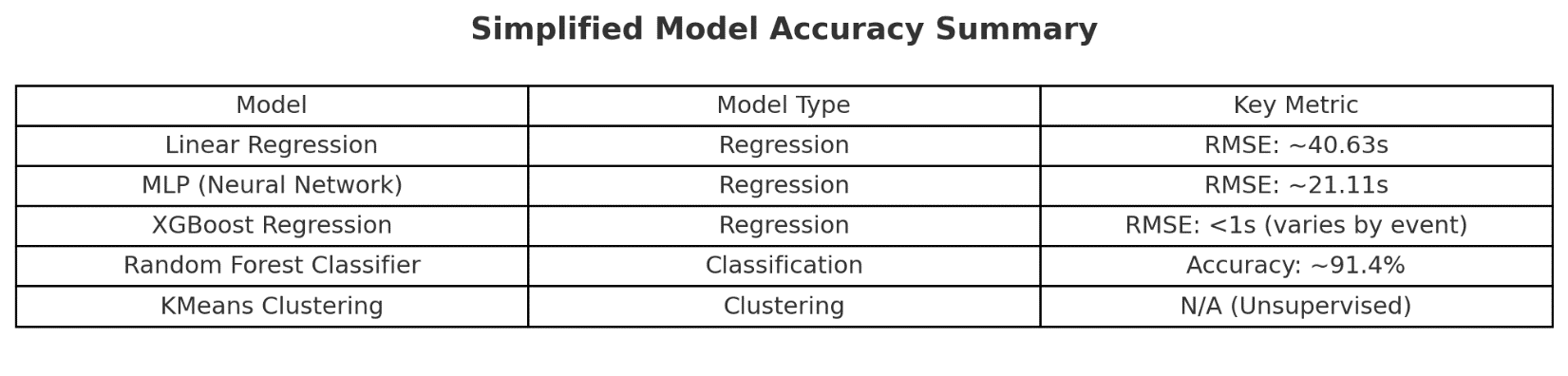


**Figure – Comparison of Linear Regression & Multilayer Perceptron**

The bar chart above compares the Test RMSE values for the two primary regression models with consistent, event-agnostic error metrics: Linear Regression and Multilayer Perceptron. The Linear Regression model had the highest error, with an RMSE of approximately 40.63 seconds, highlighting its limited capacity to model complex patterns in swimmer performance.

The MLP model significantly reduced this error to around 21.11 seconds, benefiting from its nonlinear architecture and fine-tuned hyperparameters.

Although not included in the bar chart due to its event-specific RMSE outputs, the XGBoost Regression model consistently outperformed both, achieving RMSE values under 1 second for most events, with R² scores approaching 1.0. Its superior accuracy and interpretability (via SHAP and PDP) make it the most effective model used in this study.



**Figure - Comparative Summary of Machine Learning Model Performance**

The table above summarizes the key performance metrics across all models used in the project. Among the regression models, XGBoost achieved the highest accuracy, with RMSE values under 1 second for most events. The MLP neural network showed strong performance with an RMSE of ~21.11 seconds, significantly improving upon the Linear Regression baseline (RMSE ~40.63 seconds).

The Random Forest Classifier accurately predicted performance categories (Elite, Competitive, Developing) with an accuracy of approximately 91.4%, while the KMeans Clustering model offered useful unsupervised grouping of athletes, though it lacked formal accuracy metrics. Overall, these results provide a comprehensive framework for analyzing and predicting swimming performance through both supervised and unsupervised learning approaches.

# 5.0 Evaluation and Analysis

This project set out to develop and evaluate various machine learning models to predict competitive swimming performance and uncover deeper patterns within the dataset. Through a structured approach — starting from baseline models to more advanced architectures — a clear progression in model accuracy, interpretability, and practicality was observed.

## 5.1 Model Comparison and Insights

The Linear Regression model served as a foundational benchmark. While it offered a quick, interpretable approach, it fell short in predictive accuracy. Its high RMSE (~40.63 seconds) and modest R² score indicated that linear patterns alone were insufficient to explain performance, especially in a sport like swimming where the interaction between distance, ranking, and event type is often nonlinear. The residual plots and prediction errors further confirmed that the model struggled particularly with longer-distance events.

The Multilayer Perceptron (MLP) model brought in the ability to learn more complex patterns that a simple linear model could not. After trying different settings and tuning the model — including adding more layers (8 and 4 neurons), allowing the model to stop early if it stopped improving, and increasing the number of training rounds — the MLP performed much better than the basic linear model. It achieved a test RMSE of around 21.11 seconds, and an R² score of 0.99165, which means it could explain most of the differences in swim times. However, when the model was tested multiple times using cross-validation, the results were less stable, with a CV RMSE of about 61.00 ± 54.49 seconds. This suggests that the MLP might be sensitive to how the data is split and may not perform as well in every situation.

The XGBoost Regression model gave the best results overall. It consistently outperformed all the other models in terms of accuracy. For most short- and mid-distance events, it predicted swim times with an error of less than 1 second. It also achieved R² scores close to 1.00, which means it was very accurate at explaining differences in swim times. Even for longer races like the 1500m freestyle, its performance stayed strong.

What made XGBoost even more impressive was its ability to explain how it made predictions. Using tools like SHAP (SHapley Additive Explanations) and PDP (Partial Dependence Plot), we were able to see which features mattered most. These tools showed that:

* Distance had the biggest impact; longer races led to longer swim times, as expected.
* Ranking was also important swimmers with better rankings usually had faster times.

Overall, XGBoost was the best-performing and most understandable model in this project. It combined high accuracy, reliable performance across different events, and clear explanations, making it the most valuable tool for predicting swim times.

## 5.2 Clustering and Classification

In addition to regression, unsupervised clustering was used to identify natural groupings among swimmers based on pace, time, and distance. The KMeans model effectively categorized swimmers into three performance-based clusters:

* Cluster 0 (Fastest): Short-distance elite swimmers
* Cluster 1 (Moderate): Middle-range performers
* Cluster 2 (Slowest): Long-distance or lower-ranked swimmers

Visualizations (e.g., boxplots, scatter plots) demonstrated how pace and distance shaped these clusters. Star swimmers such as Mark Spitz, Katie Ledecky, and Michael Phelps were successfully identified in the highest-performing clusters, adding a layer of validation to the clustering logic.

Building on this, a classification model was developed to predict swimmer categories:

* Elite (Top 20%)
* Competitive (Middle 60%)
* Developing (Bottom 20%)

This classification model accurately reflected the cluster-based performance tiers and allowed swimmer profiling. Such categorization could be especially useful for coaches and analysts aiming to segment athletes by development potential or competition readiness.

## 5.3 Implications and Real-World Relevance

The integration of regression, classification, and clustering in this project provides a flexible and robust framework for analyzing swimming performance. These models have significant real-world applications across various roles in competitive sports. For coaches, the classification and clustering models can support talent identification, enabling them to group swimmers by performance tier and tailor training programs based on current ability. Predictive regression models can also be used for race simulation, offering estimated finish times under different conditions.

From an analytical standpoint, the use of interpretability tools such as SHAP (SHapley Additive Explanations) and Partial Dependence Plots allows for transparent model evaluation, giving performance analysts deeper knowledge into which factors most influence race outcomes. The framework developed here could serve as a foundation for sports scientists, who may apply these techniques to track swimmer development over time or explore external influences such as altitude training, nutrition, or rest cycles.

## 5.4 Limitations

Despite the strong performance of the models, several limitations should be acknowledged. First, event imbalance was an issue certain swimming events had significantly fewer entries than others, which may have limited the generalizability of the models for those event types. To maintain model integrity, only events with at least 50 samples were included in the final regression evaluation, which meant some athlete performances were excluded altogether.

Additionally, the dataset lacked key biographical or training-related features such as swimmer age, number of years competing, training hours per week, or access to coaching staff — all of which could contribute meaningfully to predictive performance modeling. While date fields were preprocessed and converted, no time-series analysis was conducted, so we were unable to assess performance progression or trends over time.

## 5.5 Best Performing Model

Throughout this project, several models were developed and tested to predict competitive swimming performance. These included Linear Regression, Multilayer Perceptron (MLP), and XGBoost for regression tasks, as well as Random Forest for classification and K-Means for unsupervised grouping. After a thorough evaluation of all performance metrics, the XGBoost Regression model emerged as the best-performing and most reliable model in the study.

The XGBoost model consistently delivered the lowest prediction errors across nearly all events. For short-distance races such as the 50m and 100m freestyle, it achieved RMSE values below 1 second, which is extremely accurate given the short time spans of these events. Even in longer races like the 1500m freestyle, where variability is higher, the model maintained a strong performance with RMSE under 18 seconds outperforming the MLP and Linear Regression models by a significant margin.

In terms of explanatory power, XGBoost achieved R² scores near 1.00, showing that it was able to explain nearly all the variance in swim times based on features like Distance, Ranking\_numeric, Sex, and Event. It provided excellent interpretability through SHAP (SHapley Additive Explanations) and Partial Dependence Plots (PDP). These tools allowed for clear insight into how each feature influenced predictions, making the model not only powerful but also transparent.

The SHAP analysis revealed that:

* Distance was the most influential feature — as expected, longer races resulted in longer times.
* Ranking\_numeric showed that better-ranked swimmers (lower numbers) were generally faster.
* Sex and Event type had moderate influence, capturing gender-based and technical differences in race types.

Moreover, XGBoost's performance was stable and consistent across different data splits, addressing the variability concerns observed in MLP's cross-validation results. Its ability to model complex relationships, avoid overfitting, and still offer model interpretability makes it the most practical and trustworthy solution for real-world applications such as race time forecasting, swimmer benchmarking, and performance tracking.In conclusion, XGBoost combined the highest accuracy, strongest generalization, and clear interpretability, making it the best performing model in this project and the most suitable for use in competitive swimming analytics.

# 6. Project Management Approach

This section explains how I managed the project from start to finish. It includes the approach I used, how tasks were planned and scheduled, and how I tracked my progress. I also describe the timeline of the work, show how I stayed on schedule, and include a table of supervisor meetings and feedback. This helped me stay organised and complete the project on time.

## 6.1 Project Methodology

For this project, I followed the Waterfall method, which means work was completed on each stage one after the other. The work was carried out in distinct and sequential phases, starting with data collection and preprocessing, followed by model development, evaluation, and reporting. Each stage was primarily completed before progressing to the next, with limited iteration or backtracking.

Agile frameworks are designed for flexible, team-based development, where requirements may evolve and continuous feedback is available. However, this project was conducted as a solo, academic research dissertation with clearly defined goals and deadlines. The objectives were fixed from the start, such as predicting swim times and categorizing athlete performance, and did not require ongoing adaptation or stakeholder feedback throughout.

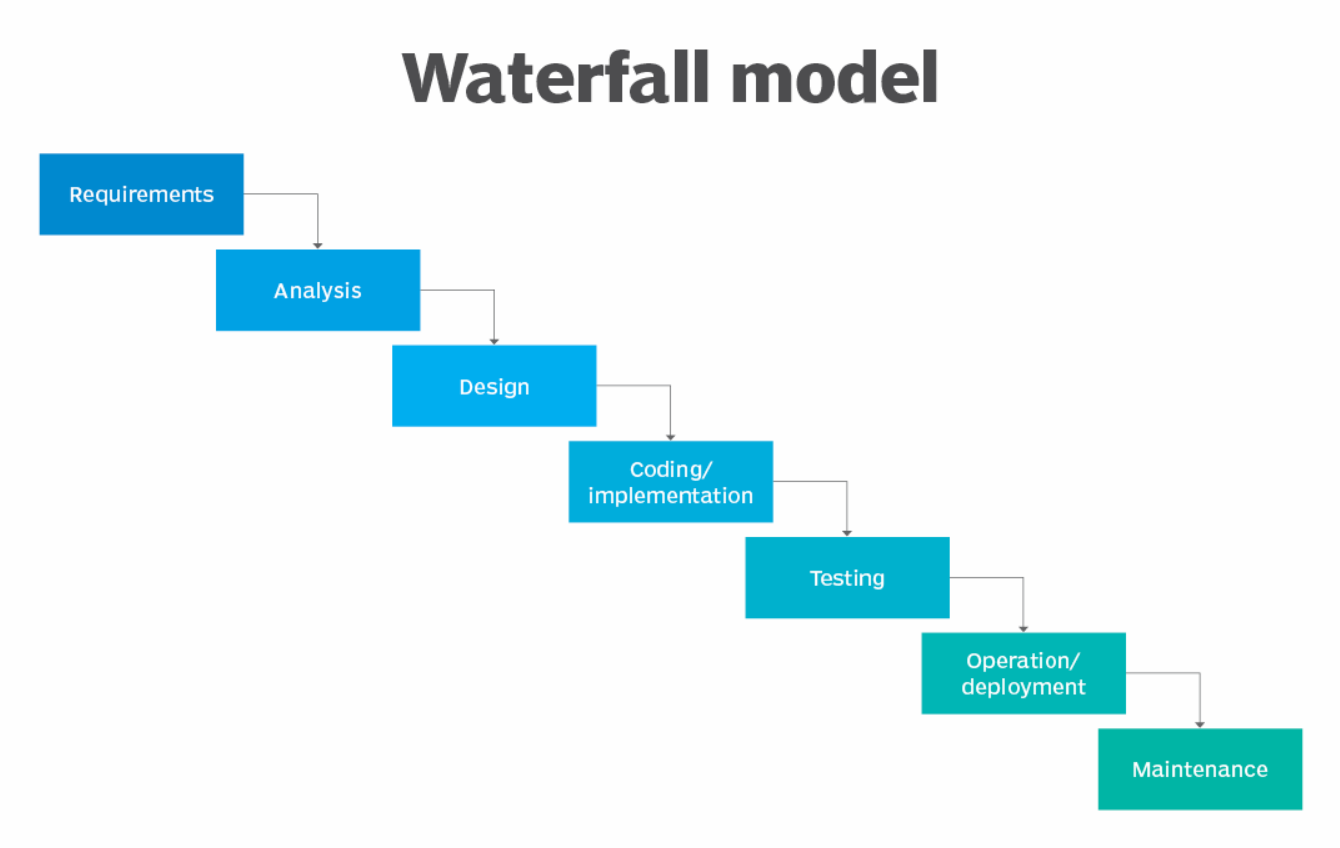
While Agile focuses on iterative development with continuous feedback loops and evolving requirements, Waterfall follows a linear, sequential approach where each phase is completed before the next begins. Agile is better suited for collaborative, fast-paced environments with changing goals, whereas Waterfall is more structured and ideal for solo academic projects with clearly defined deliverables In this context, Waterfall provided a more stable and manageable framework for progressing through each research stage with minimal disruption.

Waterfall provided a better fit for this project because it supports a structured, phase-by-phase workflow, which is ideal for research-based work. Each phase had a specific focus:

* Preprocessing the dataset
* Selecting and training models
* Evaluating results
* Documenting the findings

This linear structure helped maintain clarity and organization throughout the dissertation and ensured that the outcome was built on a solid foundation of cleaned data and methodical experimentation.

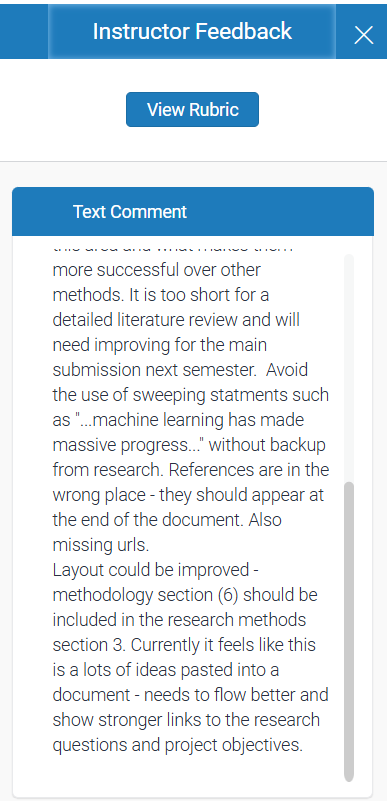
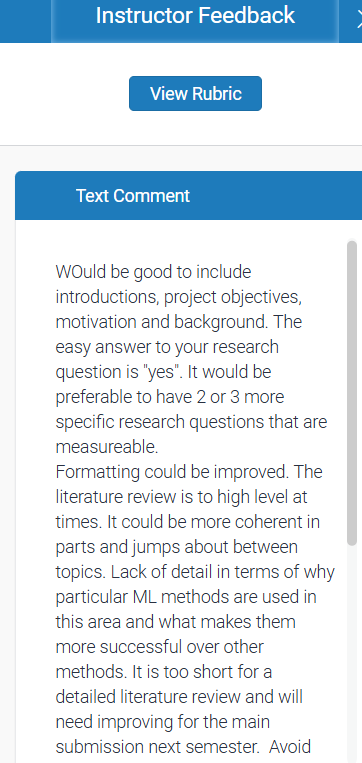
In future projects, especially those involving collaboration, evolving requirements, or iterative model improvement, I would consider applying an Agile or hybrid methodology. However, for a solo academic project with defined deliverables and a fixed scope, Waterfall provided the most practical and efficient framework.



**Figure - The Waterfall Model Development Lifecycle**

The diagram above shows the Waterfall model, which follows a step-by-step process where each stage is completed before the next one begins. This method matched the way I approached my project. I started with the requirements and analysis stage by reviewing past research, setting the aim of the study, and planning the project. Then, in the design phase, I decided which models to use and how to prepare the dataset. The implementation stage included writing code, training models like Linear Regression, MLP, and XGBoost, and adjusting them for better results. After that, I moved into the testing phase, where I compared model performance using accuracy metrics like RMSE and R². Finally, in the deployment and maintenance phase, I focused on writing up results, refining the report, and reflecting on what I learned. Following this clear structure helped me manage time well and avoid confusion between tasks.

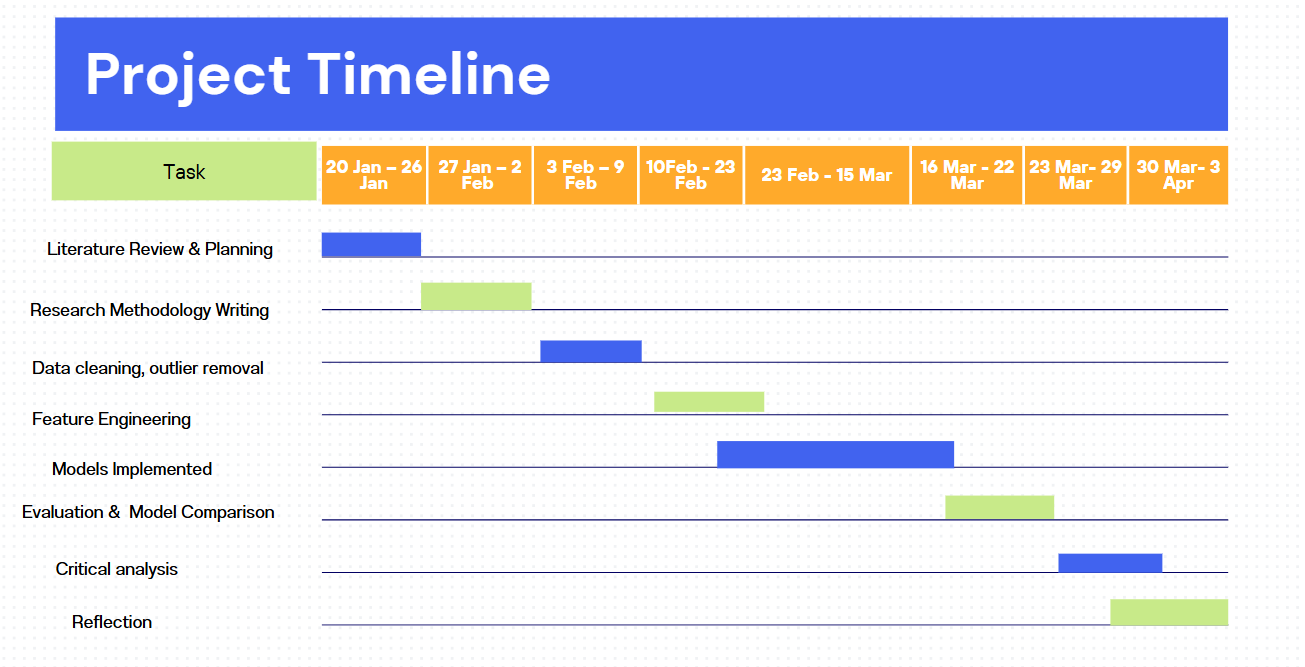
## 6.2 Supervisor Feedback

** **

The feedback I received on my initial project proposal in the previous semester in 6000CEM played an important role in shaping the final version of this dissertation. My supervisor advised me to improve the structure of the report by including a clear introduction, project objectives, and background context. I made sure to address this by expanding the Introduction and Literature Review sections, clearly stating the aim and refining the central research question into a more specific, measurable form. The comments also highlighted that the earlier literature review lacked coherence, depth, and justification for the chosen methods. In response, I reviewed a wider range of studies and restructured the literature review to better explain why certain machine learning models, such as XGBoost and MLP, are suitable for performance prediction in swimming. This early feedback helped guide the research process and ensured that the final report met academic standards in both content and structure.

## 6.3 Gantt Chart Timeline

To stay organized, I created a weekly task plan and used a Gantt chart to visualize it. This chart helped me allocate enough time for each phase and ensured I didn’t leave writing or model evaluation until the last minute.



**Figure - Gantt Chart Representing Project Timeline**

The Gantt chart breaks down the entire dissertation into weekly phases, starting from 20th January to 3rd April. The chart visually maps key tasks such as literature review, data preprocessing, model development, and final reflection, showing how each phase was distributed over time. This structured timeline helped ensure that all stages were completed in a logical and organized manner, avoiding a last-minute rush and allowing time for evaluation and refinement.

## 6.4 Supervisor Meetings

I met with my supervisor regularly for guidance and progress reviews. These meetings helped shape the project’s direction, especially around modeling decisions and reporting. The table below shows when the meetings occurred and what was discussed.

| **Dates** | **Task** | **Supervisor Meeting** | **Notes** |
| --- | --- | --- | --- |
| 20 Jan – 26 Jan | Literature Review & Planning | Yes | Initial discussion, confirmed project direction, and tools to be used |
| 27 Jan – 2 Feb | Research Methodology Writing | No | Continued to work independently, writing basic methods and a model plan |
| 3 Feb – 9 Feb | Data Cleaning & Outlier Removal | Yes | Reviewed data sources, discussed and method for cleaning |
| 10 Feb – 23 Feb | Feature Engineering | No | Applied coding and feature selection, and progressed based on the earlier plan |
| 23 Feb – 15 Mar | Models Implemented (LR, MLP, XGBoost) | Yes | Reviewed model performance, shifted focus to XGBoost due to better results |
| 16 Mar – 22 Mar | Evaluation & Model Comparison | Yes | Compared models, introduced data visualisation for explainability |
| 23 Mar – 29 Mar | Critical Analysis | No | Wrote independently, reviewed limitations and insights |
| 30 Mar – 3 Apr | Reflection + Final Touches | Yes | Final review before submission, received feedback on structure and polish |

The table below presents a summary of the supervisor meetings held during the project. These meetings took place at key points throughout the development process and provided valuable feedback and guidance. Early meetings focused on confirming the project direction and data sources, while later sessions helped with improving the methodology, evaluating model performance, and refining the structure of the report. In each meeting, I discussed my progress, asked questions, and received suggestions that helped me solve technical challenges and stay on track. This regular communication played an important role in keeping the project organised and aligned with academic expectations.

**6.**5 Reflection on Time Management and Challenges

The project timeline was generally followed as planned, thanks to early scheduling and a structured Gantt chart that guided weekly progress. Most tasks, including data cleaning, model development, and writing, were completed within their target weeks. To manage my time effectively, I worked on a weekly checklist using Google Sheets and ensured each task was broken into smaller steps, which made the workload feel more manageable.

Some challenges did arise during the project. For example, the MLP neural network initially failed to converge and required extra tuning and adjustments. To avoid delays, I shifted some writing tasks earlier and extended technical work into the next phase. During data cleaning, I encountered unusually high swim time entries that required the use of IQR-based outlier detection, an unplanned but essential task. These issues were resolved by adapting the timeline and maintaining flexibility in overlapping tasks like model evaluation and report writing. Despite these minor changes, the overall timeline remained on time, and the project was completed on schedule. Effective planning and the ability to adjust when needed were key in keeping everything on track.

In summary, following a clear plan and timeline helped me stay focused and avoid last-minute delays. When challenges came up, I adjusted my schedule and kept moving forward. Regular supervisor feedback, weekly goals, and time management tools were key in making sure each part of the project was completed successfully and on time.

# 7. Conclusion

## 7.2 Introduction

This project explored whether machine learning could be used to predict competitive swimming performance and identify key factors that influence race outcomes. The main goal was to apply different machine learning models to a historical swimming dataset and test how well each model could predict swim times and classify swimmers based on their performance levels. By using regression, classification, and clustering methods, the study aimed to create a complete picture of swimmer performance, offering insights that could support talent identification and training strategies in real-world sports environments.

## 7.3 Achievements

The project successfully achieved its objectives through a series of well-defined steps. First, the raw dataset was carefully cleaned and prepared, including handling missing values, converting race times into numerical formats, and removing outliers using the Interquartile Range (IQR) method. These steps were important to ensure the data used for training was reliable and realistic.

Several machine learning models were developed and tested. Linear Regression provided a baseline, but it showed limited accuracy. The Multilayer Perceptron neural network improved prediction accuracy by capturing nonlinear patterns, although it showed variability in cross-validation. The best-performing model was XGBoost, which delivered highly accurate predictions with very low RMSE values, often under one second and R² scores close to 1.0 across most race events. This made it the most suitable model for predicting race times.

In addition to regression, a Random Forest classifier was used to group swimmers into three categories: Elite, Competitive, and Developing. This classification was based on percentile rankings and helped reveal useful patterns in athlete performance. The project also used K-Means clustering to group swimmers based on performance pace. These clusters were validated by the inclusion of world-renowned athletes like Michael Phelps and Katie Ledecky, which showed the model’s ability to reflect real-world competitive tiers.

Explainability tools like SHAP and Partial Dependence Plots added another layer of value by showing which features most affected predictions. Distance was consistently the most influential factor, followed by swimmer ranking. These insights help coaches understand not only what the models predict, but also why.

## 7.4 Future Development

While the results were promising, there are several ways this project could be improved or expanded in the future. One improvement would be to include more detailed data about each swimmer, such as age, training volume, or health information. These additional features could improve the accuracy and depth of predictions, especially when trying to forecast long-term performance or training needs.

Although LSTM (Long Short-Term Memory) networks are powerful for time-series prediction tasks, they were not implemented in this project due to the structure of the available dataset. LSTM models require sequential data for each individual, meaning several race performances from the same swimmer over time. However, the dataset used in this study included mainly single-event entries or very few records per swimmer, which made it unsuitable for training a model that depends on understanding how performance changes over time. Without enough time-based data per athlete, an LSTM would not be able to learn meaningful patterns or trends.

In future work, if a longitudinal dataset is collected, with multiple performance records per swimmer over months or years, then LSTM models could be very effective. They could help track individual progress, identify performance plateaus, and even forecast peak performance windows. This would make the prediction more personal and dynamic, giving coaches insights not just into where an athlete stands now, but where they are heading.

Finally, building an interactive dashboard or web application could make this system more accessible for coaches and swimmers. By allowing users to input swimmer data and receive instant predictions or insights, the project could move beyond research and into real-world use. This could support performance tracking, race simulation, and personalized training plans, especially in high-performance environments where small gains can make a big difference.

# 8. Reflection

This project has been a major learning experience for me, both technically and personally. When I started, I had a basic understanding of machine learning and data preprocessing, but through this project, I was able to apply that knowledge in a real world scenario. I improved my skills in working with messy datasets, handling missing values, removing outliers, and preparing features for modeling. I also became much more confident using libraries like scikit-learn, XGBoost, and SHAP, which I had never worked with in depth before.

One of the biggest technical challenges I faced was building and tuning the MLP neural network. At first, the model didn't perform well and failed to converge properly. I had to spend extra time researching solutions, adjusting the architecture, and using tools like GridSearchCV and early stopping. This process taught me how much trial and error is involved in machine learning and how small changes in parameters can make a big difference in results.

Time management was another important skill I developed. I used a Gantt chart and a weekly checklist to stay on track, and I learned the importance of leaving buffer time for unexpected problems, which helped when tasks like model tuning or SHAP analysis took longer than expected. Having regular supervisor meetings also kept me focused and gave me helpful direction whenever I felt stuck.

If I were to do this project again, I would try to collect or use a dataset with more time-based records per swimmer. That way, I could apply models like LSTM to analyze how performance changes over time. I would also aim to improve my visual storytelling skills so I could present my results in a more interactive or dashboard format.

Overall, this project has helped me grow as a data science student. I now have hands-on experience applying different machine learning models, evaluating them, and explaining their outputs clearly. I also feel more confident in my ability to manage a long-term project from start to finish — including planning, coding, analyzing, and writing. Most importantly, I’ve learned how to keep going when things don’t work right away, and how to turn feedback and mistakes into improvements.

# 9. References

Allen, S. v., Vandenbogaerde, T. J., & Hopkins, W. G. (2014). Career performance trajectories of Olympic swimmers: Benchmarks for talent development. *European Journal of Sport Science*, *14*(7), 643–651. <https://doi.org/10.1080/17461391.2014.893020>

Maszczyk, A., Roczniok, R., Waśkiewicz, Z., Czuba, M., Mikolajec, K., Zajac, A., & Stanula, A. (2012). Application of regression and neural models to predict competitive swimming performance. *Perceptual and Motor Skills*, *114*(2), 610–626. <https://doi.org/10.2466/05.10.PMS.114.2.610-626>

Stanula, A., Maszczyk, A., Roczniok, R., Pietraszewski, P., Ostrowski, A., Zajac, A., & Strzała, M. (2012). The development and prediction of athletic performance in freestyle swimming. *Journal of Human Kinetics*, *32*(1), 97–107. <https://doi.org/10.2478/v10078-012-0027-3>

Wilk, R., Fidos-Czuba, O., Rutkowski, Ł., Kozłowski, K., Wiśniewski, P., Maszczyk, A., Stanula, A., & Roczniok, R. (2015). PrEdiCtiNg CoMPEtitiVE SwiMMiNg PErforMaNCE. In *Central European Journal of Sport Sciences and Medicine |* (Vol. 9, Issue 1).

Xie, J., Xu, J., Nie, C., & Nie, Q. (2017). Machine learning of swimming data via wisdom of crowd and regression analysis. *Mathematical Biosciences and Engineering*, *14*(2), 511–527. <https://doi.org/10.3934/mbe.2017031>

Yustres Amores, I., Santos del Cerro, J., González-Mohíno, F., Hermosilla, F., & González-Ravé, J. M. (2023). Modelling performance by continents in swimming. *Frontiers in Physiology*, *14*. <https://doi.org/10.3389/fphys.2023.1075167>

Brefeld, U., Davis, J., van Haaren, J., & Zimmermann, A. (Eds.). (2022). Machine Learning and Data Mining for Sports Analytics (Vol. 1571). Springer International Publishing. <https://doi.org/10.1007/978-3-031-02044-5>

Davis, J., Bransen, L., Devos, L., Jaspers, A., Meert, W., Robberechts, P., van Haaren, J., & van Roy, M. (2024). Methodology and evaluation in sports analytics: challenges, approaches, and lessons learned. Machine Learning, 113(9), 6977–7010. <https://doi.org/10.1007/s10994-024-06585-0>

Edelmann-Nusser, J., Hohmann, A., & Henneberg, B. (2002). Modeling and prediction of competitive performance in swimming upon neural networks. European Journal of Sport Science, 2(2), 1–10. <https://doi.org/10.1080/17461390200072201>

Jensen, U., Prade, F., & Eskofier, B. M. (2013). Classification of kinematic swimming data with emphasis on resource consumption. 2013 IEEE International Conference on Body Sensor Networks, BSN 2013. <https://doi.org/10.1109/BSN.2013.6575501>

Powell, L., Polsley, S., Casey, D., & Hammond, T. (2023). The Real-Time Classification of Competency Swimming Activity Through Machine Learning. International Journal of Aquatic Research and Education, 14(1). <https://doi.org/10.25035/ijare.14.01.06>

vaso, M., Knechtle, B., Rüst, C. A., Rosemann, T., & Lepers, R. (2013). Age of peak swim speed and sex difference in performance in medley and freestyle swimming - A comparison between 200 m and 400 m in Swiss elite swimmers. Journal of Human Sport and Exercise, 8(4), 954–965. <https://doi.org/10.4100/jhse.2013.84.06>

Worsey, M. T. O., Pahl, R., Espinosa, H. G., Shepherd, J. B., & Thiel, D. v. (2021). Is machine learning and automatic classification of swimming data what unlocks the power of inertial measurement units in swimming? Journal of Sports Sciences, 39(18), 2095–2114. <https://doi.org/10.1080/02640414.2021.1918432>

Zheng, T., & Sun, Y. (2022). A Data-Driven Analytical System to Optimize Swimming Training and Competition Performance using Machine Learning and Big Data Analysis. 95–104. <https://doi.org/10.5121/csit.2022.121508>

Wu, P. P. Y., Babaei, T., O’Shea, M., Mengersen, K., Drovandi, C., McGibbon, K. E., Pyne, D. B., Mitchell, L. J. G., & Osborne, M. A. (2021). Predicting performance in 4 x 200-m freestyle swimming relay events. PLoS ONE, 16(7 July). <https://doi.org/10.1371/journal.pone.0254538>

Louro, H., Freitas, J. D., Borrego, C., Cunha, R., Marinho, D., Santos, C., & Conceição, A. (2021). Analysis and prediction of competition performance in master swimmers. Motricidade, 17(4), 391–398. <https://doi.org/10.6063/motricidade.25521>

# 11. Appendix – Github Link

<https://github.coventry.ac.uk/abdulla23/6001CEM_MuhammadAbdullah>