B.Tech Project Report
(CSPE 40)
on

Case-based Time Prediction for Recreational Marathon Runners

Pransu Yadav (12022001)
Anurag Shrestha (12022002)
Inzamam ul Haque Chowdhury(12022004)

Under the Supervision of
Dr. Vijay Verma
Asst. Prof.

Department of Computer Engineering
National Institute of Technology, Kurukshetra
Haryana-136119, India
Jan-May 2024

Certificate

We, hereby certify that the work which is being presented in this B.Tech Project (CSPE40) report entitled "***Case-based Time Prediction for Recreational Marathon Runners*** ", in partial fulfillment of the requirements for the **Bachelor of Technology in Computer Engineering** is an authentic record of our own work carried out during a period from January, 2024 to May, 2024 under the supervision of **Dr. Vijay Verma**, Assistant Professor, Computer Engineering Department.

The matter presented in this project report has not been submitted for the award of any other degree elsewhere.

Signature of Candidate

Pransu Yadav (12022001)

Anurag Shrestha (12022002)

Inzamam ul Haque Chowdhury(12022004)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Signature of Supervisor Faculty Mentor
Dr. Vijay Verma
Asst. Prof.

Table of Contents

Abstract

Exercise is widely recognized as being important for maintaining a healthy lifestyle in today's fitness-conscious world. Running, cycling, and triathlons have all become very popular as ways to push oneself and keep in shape. Nevertheless, many recreational athletes find themselves navigating through generic or slightly tailored training programs, which frequently fail to appropriately encourage newcomers on their fitness goals, despite the abundance of training advice available. To close this gap, we suggest a unique method that applies case-based reasoning to provide individualized time prediction for marathon runners. Our method leverages the abundance of activity data that is regularly collected by smartwatches and apps such as Strava. It compares the training histories of individual runners with those of similar runners who have similar race goals.

Our approach is innovative in that it uses prefactual explanations to help runners modify their training plans as their fitness objectives change over time. We want to give runners individualized support during their preparation so they feel more motivated and committed on race day. We do this by having a conversation with them about their goals and progress. We show that it is possible to generate individualized time predictions for up to 80% of real-world runners through evaluation utilizing a huge dataset that includes over 800 runners. Interestingly, the suggestions made are not only doable but also useful when incorporated into a runner's training regimen. These usually entail little, gradual changes to factors like weekly mileage, long-run time, or average training pace.

Our strategy essentially marks a paradigm shift in the way marathon runners receive training guidance, moving away from general time predictions and toward tailored assistance based on contextual knowledge and empirical data. We aim to create a more welcoming and encouraging atmosphere for new athletes by providing them with customized advice based on their individual goals and profiles. This will help them on their path to accomplishing their race day objectives.

1. Introduction

The use of user modeling and personalization approaches (UMAP) to assist recreational marathon runners with their training and competitiveness is examined in this study. There's been a rising interest in extending UMAP methodologies to real-world activities like marathon running, even though the study has generally focused on online behaviors like product preferences or learning engagement. Because of its highly driven participants—many of whom are novices in need of assistance—and the range of characteristics of marathon preparation that lend themselves to individualized advice, the marathon is an appealing subject for personalization research.

Marathon training usually takes 12–16 weeks, although recreational runners don't always have access to individualized coaching. Current online training regimens are not very customizable or flexible enough to accommodate individual demands or unforeseen events like accidents or hectic schedules. Individualized training features are increasingly included in wearable technologies, such as Garmin devices, which broadens the audience for individualized training methods.

The approach detailed in the report uses data gathered from mobile devices and smartwatches during training sessions to target inexperienced marathon runners. Based on the training regimens of runners who are comparable to one another, it uses case-based reasoning (CBR)[6] to suggest training modifications. The technology forecasts a runner's race time and makes customized training time predictions based on an analysis of their past training records. This entails having a conversation with the runner about their objectives and making specific time predictions for changing their training schedule[1]. These time predictions consider the experiences of runners who are similar and are grounded in prefactual reasoning.[2,7,8]

The method was evaluated retrospectively in the research using data from almost 300,000 recreational runners who trained for 500,000 marathons on Strava between 2014 and 2017. This assessment shows the potential of UMAP approaches in helping recreational marathon runners by evaluating how successful the individualized training time predictions are.

2. Motivation

This project's impetus comes from the realization of how important exercise is to upholding a healthy lifestyle. People are becoming more conscious of the value of physical activity as sports like cycling, triathlons, and running become more popular as ways to keep in shape and push themselves.But even with all the training guides and plans out there, recreational athletes—especially first-time runners—find themselves lacking in support as they pursue their fitness goals. Current training guidelines frequently provide general or very loosely customized advice, failing to take into account the various demands and objectives of individual athletes. This lack of individualized support may deter runners from sticking to their fitness objectives and result in less than ideal training results.

The research feels the gap by introducing a case-based reasoning system that can produce training time predictions that are unique to each recreational marathon runner. The approach attempts to deliver runners personalized advice based on their training history and the experiences of the similar runners with similar race goals by utilizing the abundance of activity data typically collected by smartwatches and fitness applications like Strava. The main goal of this study is to overcome the shortcomings of one-size-fits-all training plans and offer beginner marathon runners individualized assistance during their training process. The approach aims to improve runners' training experience, maintain their motivation, and eventually increase their chances of success on race day by facilitating a discourse regarding their training progress and race goals. The research shows that it is feasible and effective to generate tailored training time predictions for a considerable fraction of runners through a thorough review utilizing real-world data from a large cohort of leisure runners. Through establishing a connection between general training plans and the unique requirements of each athlete, this study seeks to empower new runners and enhance their general health and well-being.

3. Related Work

The present culture of fitness data began a long time ago.The fitness data revolution began with the introduction of a wireless heart rate monitor in 1982 by a Finnish manufacturer , Polar[3]. People only took notice once it was brought to a more accessible market through Fitbit with cheap, wearable sensors to the public[4,5]. Today millions of people use these devices to track their daily lifestyles. Applications like Strava, Nike Running Club, have allowed its users to share their activities publicly and create a social construct around it.

In 1982, a Finnish manufacturer of sports named Polar[1] training computers, introduced the first wireless heart rate monitor which brought a fitness data revolution but this technology came into mainstream a decade later when companies like Fitbit made cheap, wearable sensors to the masses. Today, these devices are used by millions of people to track their daily activities[2,3] with apps like Strava and Runkeeper. Modern companies such as whoop, have introduced technology to integrate the data about sleep, recovery and exercise to help users to train efficiently. Services such as Training Peaks provide their users with functional estimates of their fitness to guide their training efforts and monitor their recovery. Users of Strava can receive recommendations for recuperation exercises to assist maintain their health while increasing their training efforts in advance of a major event, as well as information regarding their race-readiness. Although the market for fitness devices and applications has grown dramatically, the options available today barely scratch the surface of what may be. Machine learning and recommender system researchers have started to concentrate their attention in areas like fitness assessment, training load estimation, recovery guidance, personalized training recommendations, performance prediction, and race planning, drawn by the availability of data and a motivated user base. On the other hand, the users of Strava have access to the readiness of their race and can be recommended recovery activities to help keep them healthy as they increase their training efforts in preparation for the main race. New market opportunities are opened due to these apps but only to the scratch level.

4. Framework

4.1 Conceptual Design Diagram

As seen in Fig 4.1.1 & Fig 4.1.2, dfd for both levels is presented respectively.

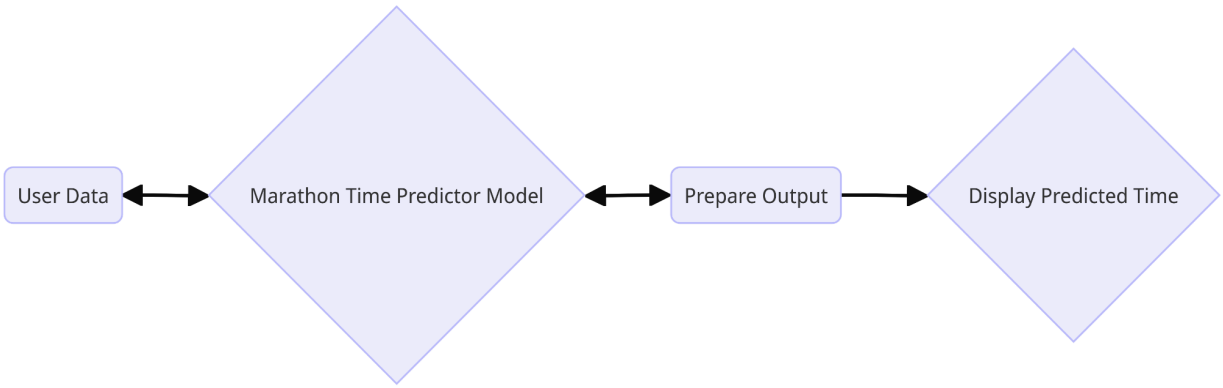

Fig:- 4.1.1 Level 0 DFD Diagram

Elements of the diagram:

- User Data: Data collected from the CSV file containing details about different marathon runners.
- Marathon Time Predictor Model: Case based Machine Learning Model that predicts the time it will take for the runner to complete the marathon.
- Prepare Output: The model will provide us the desired output that we can provide to the user.
- Display Predicted Time: Display the output time with visualized model statistics.

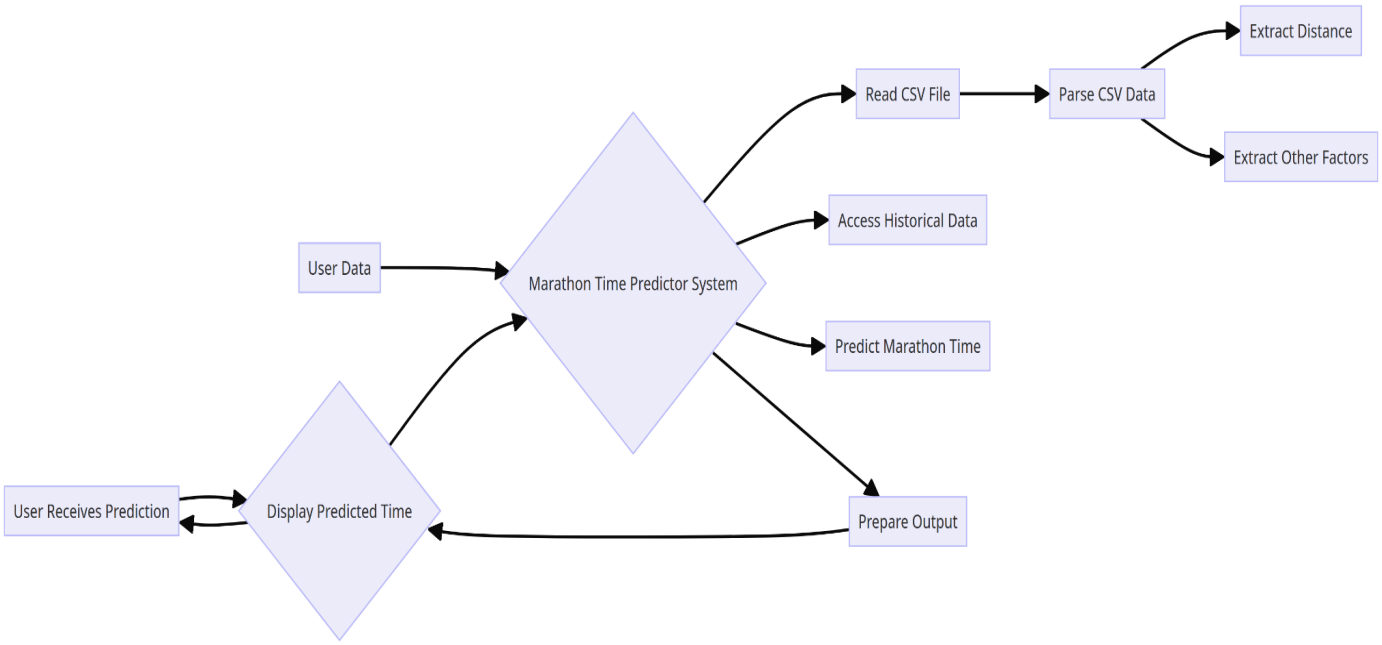

Fig:- 4.1.2 Level 1 DFD Diagram

Elements of the diagram:

- User Data: Data collected from the CSV file containing details about different marathon runners.
- Marathon Time Predictor Model: Case based Machine Learning Model that predicts the time it will take for the runner to complete the marathon.
- Prepare Output: The model will provide us the desired output that we can provide to the user.
- Display Predicted Time: Display the output time with visualized model statistics.
- Read CSV file: Model will read the csv file from input.
- Parse CSV Data: The model with do some exploratory data analysis
- Extract Distance: Model will study the distance for specific runners based on their training plans.
- Access Historical Data: Based on the data of training, the model will study the previous mentioned data.
- User receives prediction: Output is received by the user.

4.2 Algorithm

Implementing the below algorithm given in the research paper and showing results in Fig 4.2.1 & Fig 4.2.2.

Algorithm 1 is designed to generate time predictions of the recreational runners and perform pre training time estimation and training recommendation. It takes time depending on the query of the particular operator (q) and the casebase (CB). In addition, it takes into account parameters such as the current training week (w), the number of actual and preliminary data to be received, a factor that represents the difference between the runner's estimate and the expected completion time (δ). K is the total case. and significance level (p). Results include estimated marathon time (P), accuracy problem (δ), a set of preconditions (cp & cf), and accuracy and a differences in the cases (sig).

1: $C \leftarrow filter(CB, week = q.week, sex = q.sex)$
2: $C' \leftarrow sort(C, sim(q, c))$
3: $P \leftarrow mean(C'.MT.head(k))$
4: if $\delta \leq 0$ then
5: $Cf \leftarrow C'[C'.MT \geq P].head(k)$
6: $Cp \leftarrow C'[C' \leq P * (1 + \delta)].head(k)$
7: else
8: $Cf \leftarrow C'[C'.MT \leq P].head(k)$
9: $Cp \leftarrow C'[C' \geq P * (1 + \delta)].head(k)$
10: end if
11: $sig \leftarrow []$
12: for f in $C'.F'$ do
13: $sig.append(f)$ if $ttest(Cf.f, Cp.f) < p$
14: end for
15: return P, Cf, Cp, sig

Implementing the above algorithm we get the following results.

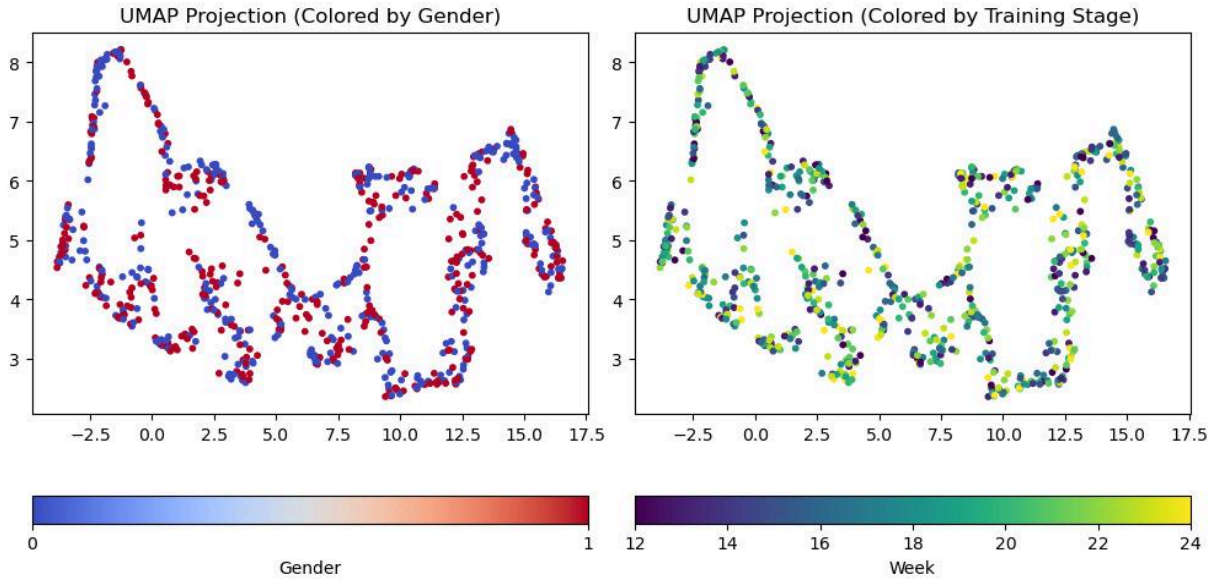

Fig:- 4.2.1 UMAP Projection

```
Predicted Marathon Time: 5.2
Factual Cases:     Runner_Id  Week  TotDist  MaxDist  NumRest  MaxBreak  MeanPace min/km  \
31     RUN771    20    62.02    31.28        4         3              6.4
43     RUN372    20    57.43    26.82        1         1              6.3
72     RUN011    20   115.70    29.84        1         3              7.2
184    RUN276    20   114.31    33.06        3         2              6.3
197    RUN194    20    41.66    21.96        3         1              7.5

     Fastest10kmPace  CumMeanTotDist  cumBestFastest10kmPace  MarathonTime  \
31              11.4           28.05                   14.13             4
43               8.9           30.15                   13.16             5
72               9.7           25.40                   14.33             5
184              7.7           26.30                   15.51             5
197              9.8           28.70                   10.17             4

    Gender
31    Male
43    Male
72    Male
184   Male
197   Male
Prefactual Cases:     Runner_Id  Week  TotDist  MaxDist  NumRest  MaxBreak  MeanPace min/km  \
35     RUN239    20    99.27    33.88        4         2              6.9
77     RUN995    20    37.18    15.93        1         1              7.9
119    RUN919    20    87.60    12.90        2         1              7.8
164    RUN021    20    66.80    25.60        4         3              8.1
313    RUN591    20    68.47    15.12        1         2              6.5

     Fastest10kmPace  CumMeanTotDist  cumBestFastest10kmPace  MarathonTime  \
35              10.8           26.65                   14.14             6
77              10.4           26.50                   11.25             6
119              9.9           27.60                   10.11             6
164              8.5           25.50                   11.48             6
313              7.1           30.30                   15.50             6

    Gender
35    Male
77    Male
119   Male
164   Male
313   Male
```

Fig:- 4.2.2 Final results of Time Prediction

5. Experimental Setup

5.1 Jupyter Notebook

The Jupyter Notebook application is an application that allows data collection to be edited in server and processed through a web browser. Jupyter Notebook applications can be run on a local desktop that does not require network access (as described in this document), or can be installed remotely to manage and access the network.

Notebooks include a "control panel" (Notebook Control Panel) in the Jupyter Notebook application that shows local documents and allows them to open or close the notebook.

5.2 Dataset Explanation

Table 5.2.1 Strava Dataset

	Runner_Id	Week	TotDist	MaxDist	NumRest	MaxBreak	MeanPace min/km	Fastest10kmPace	CumMeanTotDist	cumBestFastest10kmPace	MarathonTime	Gender
0	RUN316	17	94.89	29.44	3	1	6.8	10.3	25.90	10.29	5	Female
1	RUN854	16	26.57	39.92	3	3	6.3	7.5	27.00	10.34	5	Male
2	RUN650	13	72.41	18.50	2	1	6.1	8.1	27.40	10.59	4	Male
3	RUN915	21	107.79	22.38	2	3	8.0	9.0	28.85	15.04	4	Male
4	RUN409	12	60.53	38.88	3	1	6.3	7.2	26.80	13.29	6	Male

The above table 5.2.1 shows Strava_Dataset that has been generated using publicly available data.

Model Used: ● UMAP Evaluation Metric: ● Mean Squared Error Output data: ● plot_umap(data)	Features Used: ● C (Case Base): Filtered cases from the dataset. ● P : Predicted Marathon Time. ● Cf : Factual Cases ● Cp: Prefactual Case ● sig : Difference between factual and prefactual cases

Table 5.2.2 Berlin Marathon Dataset

	Country	Gender	Age	Time	Start Time
0	KEN	M	30	123.533333	0.00
1	ETH	M	35	123.766667	0.00
2	ETH	M	35	126.150000	0.05
3	KEN	M	30	126.216667	0.00
4	KEN	M	30	126.233333	0.00

The above table 5.2.2 shows the Berlin Marathon Dataset that has been fetched from available data of the Berlin Marathon.

Model Used: ● Random Forest Classifier ● Ridge Regression ● Dummy Regression Evaluation Metric: ● Mean Squared Error ● Mean Absolute Error	Features Used: ● le_gender: Label Encoder ● dummy: Dummy Regressor ● reg: Ridge Regression ● standardScalerX : StandardScaler()

Table 5.2.3 All Races Dataset

	place	age_class	place_in_class	bib	name	sex	nation	team	official_time	net_time	birth_date	event	event_year	distance
0	1	M20	1	1	Rui Pedro Silva	M	PT	Individual	0 days 00:29:56.000000000	0 days 00:29:53.000000000	01/01/1992	dia-do-pai	2012	10
1	2	M20	2	184	Paulo Gomes	M	PT	Individual	0 days 00:29:58.000000000	0 days 00:29:58.000000000	01/01/1992	dia-do-pai	2012	10
2	3	M20	3	3	Bruno Albuquerque	M	PT	Sporting C P	0 days 00:30:20.000000000	0 days 00:30:18.000000000	01/01/1992	dia-do-pai	2012	10
3	4	M20	4	84	Manuel Sousa	M	PT	C I Argoncilhe	0 days 00:31:27.000000000	0 days 00:31:25.000000000	01/04/1972	dia-do-pai	2012	10
4	5	M20	5	4	Luis Mendes	M	PT	Cyclones Sanitop	0 days 00:31:46.000000000	0 days 00:31:45.000000000	01/01/1992	dia-do-pai	2012	10

The above table 5.2.3 shows the All Races Dataset that has been fetched from the results of available data of the Marathon Runners running in events in Porto, Portugal.

Model Used: ● Linear Regression ● Gradient Boosting Machine Evaluation Metric: ● Mean Absolute Error	Features Used: ● XGBRegressor

5.3 Evaluation Metrics

MSE (Mean Squared Error) and MAE (Mean Absolute Error) are two commonly used metrics to show the performance of a regression model. They quantify how close the predicted values of the model are to the actual values. Here's a brief explanation of each:

Mean Squared Error (MSE):

MSE is calculated by taking the average of the squared differences between the predicted values and the actual values.

The formula for MSE is:

When given a dataset, an estimate would be

$$\text{“}S^2 = \frac{1}{n-2}\sum_{i=1}^{n} e_i^2 = \frac{SSE}{n-2} = \frac{S_{yy} - \widehat{\beta}_1 S_{xy}}{n-2}\text{”}$$

S^2 is also called the MSE, mean squared error.

Squaring the differences creates large errors more than small errors, which can be useful depending on the context.
MSE is always non-negative, and smaller values indicate better model performance. A value of 0 indicates perfect predictions.
Mean Absolute Error (MAE):

We can calculate MAE by averaging the difference between the predicted values and the actual values.
Hence the MAE is formulated as :

$$\text{“MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{y_l} - y_i\right|\text{”}$$

Where :

$\widehat{y_i} = \ Predicted\ value\ for\ the\ i^{th}\ data\ point$

$y_i = Actual\ value\ for\ the\ i^{th}\ data\ point$

$n \ = \ number\ of\ observations$

MAE provides a more intuitive measure of error because it's in the same unit as the target variable. It's less sensitive to outliers compared to MSE.MAE is also always non-negative, and smaller values indicate better model performance. A value of 0 indicates perfect predictions.

In summary, MSE and MAE are both useful metrics for evaluating regression models, with MSE being more sensitive to large errors and MAE providing a more intuitive measure of error in the same unit as the target variable. The choice between MSE and MAE often depends on the specific requirements and characteristics of the problem at hand.

5.4 Berlin Marathon Time Prediction

5.4.1. Introduction

Marathons are a global phenomenon that attracts participants of all backgrounds and skill levels. Predicting a marathon runner's finish time can provide great information for runners, coaches, and athletes. Our aim in this project is to develop a machine learning model that will predict the completion time of the marathon based on the runner's age, gender, country and start time.

5.4.2. Hypothesis

We hypothesized that certain characteristics such as age, gender, and country have a positive relationship with time to complete the marathon. Additionally, we expect the time it takes for a competitor to cross the starting line after starting the race to be a significant predictor of finish time.

5.4.3. Tools Used

We use the scikit-learn library in Python to build machine learning models to predict marathon completion times. More specifically, we investigate ridge regression and random forest regression algorithms to improve our prediction model.

5.4.6. Implementation

We have already collected and preprocessed data from the official Berlin Marathon website, resulting in the 'Berlin_Marathon_2017_Results_Clean.txt' CSV file. This dataset contains detailed information on over 39,000 finishers, including age, gender, country, starting time, and finishing time.

In conclusion, our project demonstrates the effectiveness of machine learning in predicting marathon finishing times based on various features. By further refining our models and exploring additional datasets, we can continue to enhance their accuracy and applicability in the domain of marathon prediction and analysis.

Distplot :

sns.distplot(Results['Start Time'], bins = 60, kde=False, rug=True)
As we can see in **Fig 5.4.6.1**, an interesting pattern arises when looking at a histogram of the start times. It looks like the runners take off in 3 separate blocks, with a gap of over 10 minutes between each block.

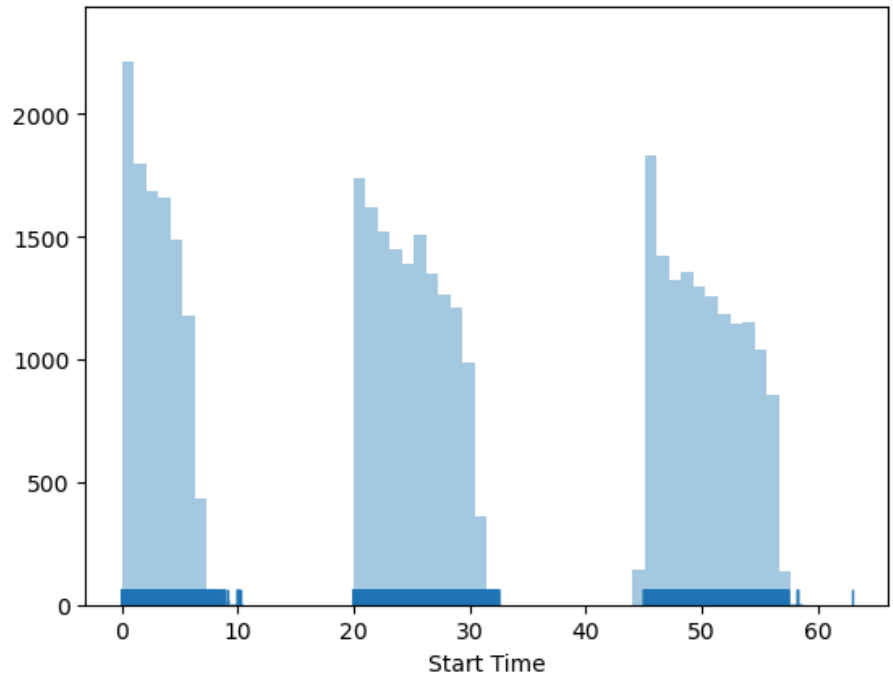

Fig 5.4.6.1 Histogram with separate blocks

To avoid too many runners and avoid collisions at the start and throughout the race, runners are often divided into groups based on finishing times, with faster runners starting at the front. However, depending on the marathon, the number of sets, the length of the set and the number of people who can cross the starting line at the same time are different. In fact, I run the same marathon every year. This means this needs to be taken into careful consideration when using the marathon or year model.

Even within the same year, it could cause issues, especially for linear models. It is unlikely to affect decision trees (and therefore random forests).

That means everyone who started in the second block (20:00 to 45:00 after the gun) will have 13 minutes subtracted from their start time (20 - 7). And everyone who started in the third block (> 45:00), will have 27 minutes subtracted from their start time ((20 - 7) + (45 - 31)).

Looking at **Fig 5.4.6.2** the histogram of adjusted start times, there are now no big gaps between the starting groups.

Adjusted distplot :

```
def adj_start(x):
    if x < 20:
        return x
    elif x < 45:
        return x - 13
    else:
        return x - 27
Results['Adjusted Start Time'] = Results['Start Time'].map(lambda x: adj_start(x))
sns.distplot(Results['Adjusted Start Time'], bins = 60, kde=False, rug=True)
```

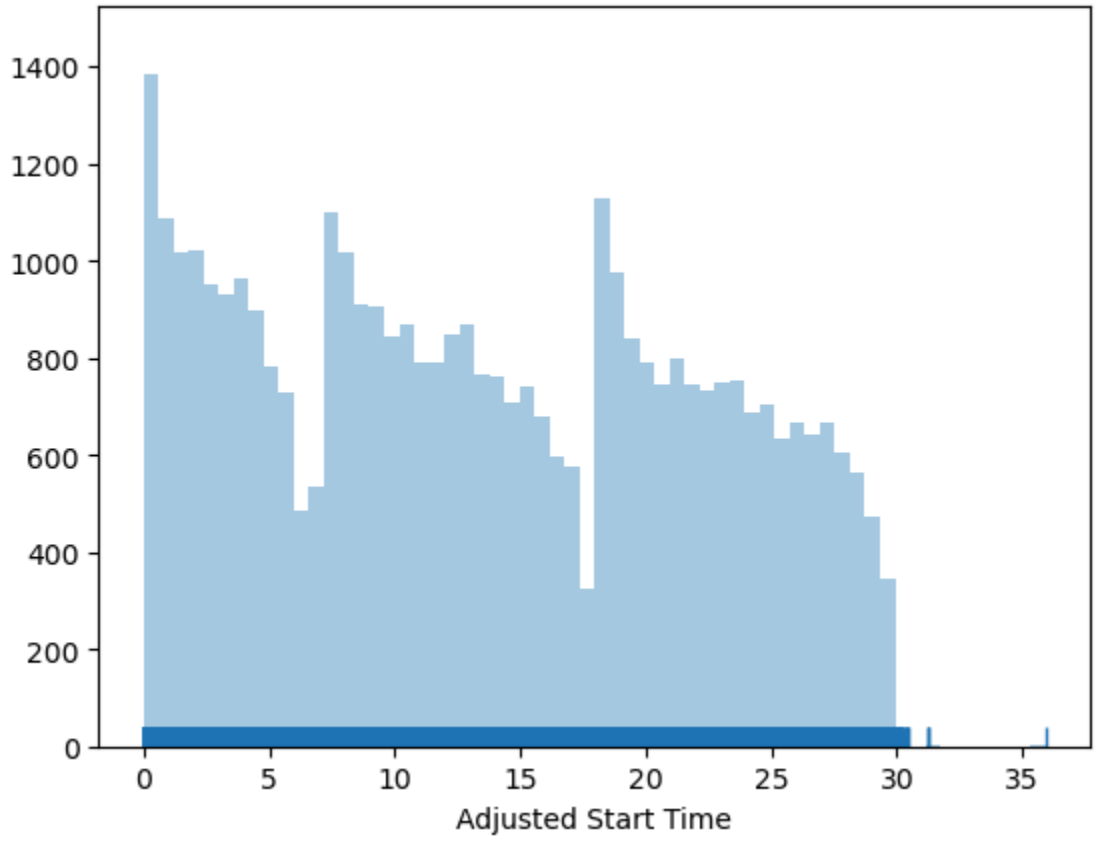

Fig 5.4.6.2:- Adjusted Histogram with no gaps.

5.4.4. Results

After training and evaluation, here are the performance metrics of our models:

Dummy Regressor :
Mean Absolute Error (MAE): [36.574535934873225]
Mean Squared Error (MSE) : [2115.2026126518654]

Ridge Regression :
Mean Absolute Error (MAE): [23.11751209759032]
Mean Squared Error (MSE) : [959.0024602923387]

Random Forest Regressor :
Mean Absolute Error (MAE): [22.50262657475109]
Mean Squared Error (MSE) : [926.7772542415825]

In conclusion, the ridge and random forest worked almost more than 50 percent better, with the random forest regression working the best.

5.4.5. Future Projects

Based on the success of our model, we identified several future studies:

Predicting 2018 Berlin Marathon results: Evaluating our model's performance in predicting 2018 Berlin Marathon results using historical data.
Berlin Marathon extension for another year:

Several factors come into play as we extend our forecast model to cover another year of the Berlin Marathon. Participants may vary from year to year due to factors such as country, age and gender working well in our prediction model. But the number of recreational runners attempting to run a marathon shows a trend that is difficult to accurately measure and correct for.

Expand to other marathons:

Covering various marathons in our model presents more difficulty. First, each marathon has a unique course and weather, resulting in significant differences in average finish times. Additionally, participants' demographic characteristics varied across countries, and our decision-making model controlled for factors such as country, age, and gender. By taking these factors into account, we aim to improve the robustness and validity of our prediction model for various marathons around the world.

5.5 Time prediction based on Half Marathon

5.5.1. Introduction

The marathon is a popular event that attracts runners of all levels, from casual gamers to elite athletes. Predicting marathon finish times is important for runners, coaches, and the athletes themselves. In this report, we present the results of a marathon running time prediction model developed using linear regression and gradient boosting machine (GBM) algorithms.

5.5.2. Dataset

We collected data from multiple marathons, including information on runner characteristics such as age, gender, training distance, and previous race time. These data contain numerical and categorical features, and the target variable is the runner's completion time. The file used is allraces.csv.

5.5.3. Methodology

We use two different learning algorithms to build predictive models:

Linear Regression: This algorithm fits a linear relationship between the input features and the target variable. It's a simple yet effective method for regression tasks.
Gradient Boosting Machine (GBM): GBM is an ensemble learning technique that builds multiple decision trees sequentially, where each tree corrects the errors of the previous one. GBM often performs well on complex datasets and can capture non-linear relationships between features and the target variable.

5.5.4. Model Training and Evaluation

We split the dataset into training and testing sets (e.g., 80% training, 20% testing) to train and evaluate our models. For both Linear Regression and GBM models, we used common evaluation metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE) to assess the models' performance on the testing set. Fig 5.5.4.1 and 5.5.4.2 show variations in Linear Regression Model. And Fig 5.5.4.3 shows Variation in GBM Model.

Linear Regression model :

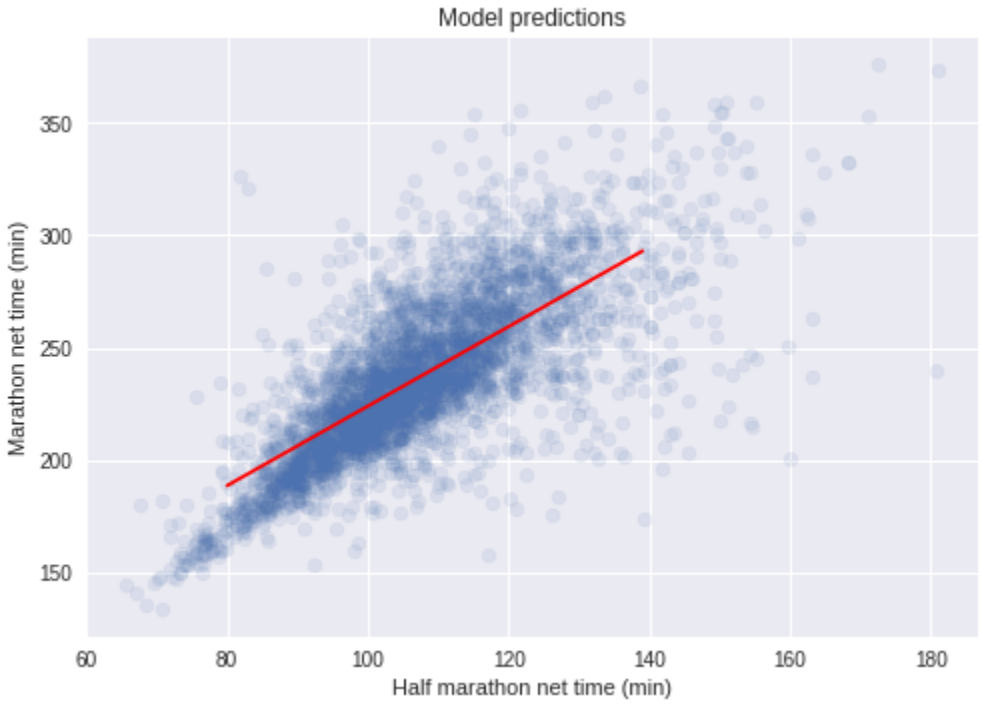

Fig :- 5.5.4.1 Linear Model

Fig 5.5.4.2 :- Linear Model with quadratic terms

GBM Model :

Fig 5.5.4.3 :- GBM Model

5.5.5. Results

After training and evaluation, here are the performance metrics of our models:

Linear Regression:

Mean Absolute Error (MAE): [17.305158988773151]

Gradient Boosting Machine (GBM):

Mean Absolute Error (MAE): [16.946433495448755]

5.5.6. Discussion

Both models performed reasonably well in predicting marathon runners' finishing times.
The GBM model outperformed the Linear Regression model, indicating that it was better able to capture the complexities in the dataset.
Further optimization and feature engineering could potentially improve the models' performance.

5.5.7. Conclusion

In summary, we develop and evaluate two marathon time estimates using linear regression and GBM algorithms. While both models gave good results, the GBM model performed better. This model could become an important tool for runners, coaches, and athletes to predict marathon completion times and improve training strategies.

5.5.8. Future Work

Explore additional features such as weather conditions, course elevation, and runner's health metrics.
Experiment with other machine learning algorithms and ensemble techniques.
Continuously update and refine the models with new data to improve prediction accuracy.
This report summarizes our efforts in building marathon runner time prediction models and provides insights into their performance and potential applications.

6. Conclusion

In conclusion, this report shows the various ways of predicting the time of marathon runners that could help them in evaluating themselves in each step of preparing for a marathon and also potentially increase their performance. Understanding a novel research paper based on an individual training plan recommendation system for recreational marathon runners, the time prediction models have been made by going through different aspects.

The method uses data collected from smartwatches and exercise apps to customize training based on the runner's goals and past learning. This personalized approach addresses the shortcomings of a one-size-fits-all training program by motivating new employees and enhancing their learning. Real-time data testing of the method has shown that it can provide good personal time estimates for many runners. We hope to support runners and improve their health and well-being through this research.

References

[1] Ciara Feely, Brian Caulfield, Aonghus Lawlor, Barry Smith. (2023) Modeling the Training Practices of Recreational Marathon Runners to Make Personalized Training Recommendations.
[2] Ruth MJ Byrne and Suzanne M Egan. 2004. Counterfactual and prefactual con-ditionals. *Canadian Journal of Experimental Psychology/Revue canadienne depsychologie expérimentale 58, 2 (2004),* 113.
[3] J Kite-Powell. 2016. Polar: The original fitness tracker and heart rate monitor. Forbes.
[4] Ranganathan Chandrasekaran, Vipanchi Katthula, Evangelos Moustakas, et al. 2020. Patterns of use and key predictors for the use of wearable healthcare devices by US adults: insights from a national survey. *Journal of medical Internet research 22, 10 (2020), e22443.*
[5] Keith M Diaz, David J Krupka, Melinda J Chang, James Peacock, Yao Ma, JeGoldsmith, Joseph E Schwartz, and Karina W Davidson. 2015. Fitbit®: An accurate and reliable device for wireless physical activity tracking. *International journal of cardiology 185 (2015), 138–140.*
[6]Ramon López de Mántaras, David McSherry, Derek G. Bridge, David B. Leake,Barry Smyth, Susan Craw, Boi Faltings, Mary Lou Maher, Michael T. Cox, Ken-neth D. Forbus, Mark T. Keane, Agnar Aamodt, and Ian D. Watson. 2005. Retrieval,reuse, revision and retention in case-based reasoning. *Knowledge Eng.Review20,3(2005),215–240. LOPEZ DE MANTARAS, R. et al. (2005) 'Retrieval, reuse, revision and retention in case-based reasoning', The Knowledge Engineering Review, 20(3), pp. 215–240. doi:10.1017/s0269888906000646.*
[7]Eoin Delaney, Derek Greene, Laurence Shalloo, Michael Lynch, and Mark T Keane. 2022. Forecasting for Sustainable Dairy Produce: Enhanced Long-Term, Milk-Supply Forecasting Using k-NN for Data Augmentation, with Prefactual Explanations for XAI. *In Case-Based Reasoning Research and Development: 30th International Conference, ICCBR 2022, Nancy, France, September 12–15, 2022, Pro-ceedings. Springer, Springer, Berlin, Heidelberg, 365–379.*
[8]Kai Epstude, Annika Scholl, and Neal J Roese. 2016. Prefactual thoughts: Mental simulations about what might happen. *Review of General Psychology 20, 1 (2016), 48–56.*
[9]Khan, S. (2023) 'Strava Dataset'**.**
[10] *Mean square error (MSE): Machine learning glossary: Encord* (2009) *Encord*. Available at: https://encord.com/glossary/mean-square-error-mse/ (Accessed: 27 April 2024).
[11] David (2021) *Berlin marathon data*, *kaggle*. Available at: https://www.kaggle.com/datasets/aiaiaidavid/berlin-marathons-data (Accessed: 27 April 2024).
[12]*Results archive* (2024) *MARATHON*. Available at: https://www.bmw-berlin-marathon.com/en/impressions/statistics-and-history/results-archive/ (Accessed: 27 April 2024).

APPENDIX

Classification of the Dataset for time prediction.

Implementation of algorithm given in the research paper:

```
def generate_predictions(q, CB, w, delta, k, p):
    # Step 1: Filter cases from the case base
    C = CB[(CB['Week'] == w) & (CB['Gender'] == q['Gender'])]

    # Step 2: Sort cases by similarity to the query case
    C_sorted = sort_by_similarity(C, q)

    # Step 3: Calculate predicted marathon time
    P = C_sorted['MarathonTime'].head(k).mean()

    # Step 4: Determine factual and prefactual cases
    if delta <= 0:
        Cf = C_sorted[C_sorted['MarathonTime'] >= P].head(k)
        Cp = C_sorted[C_sorted['MarathonTime'] <= P * (1 + delta)].head(k)
    else:
        Cf = C_sorted[C_sorted['MarathonTime'] <= P].head(k)
        Cp = C_sorted[C_sorted['MarathonTime'] >= P * (1 + delta)].head(k)
```

Dummy Regression Analysis and determining the evaluation metrics

```
dummy = DummyRegressor()
dummy.fit(X_train, y_train)
y_test_dummy_pred = dummy.predict(X_test)
mean_squared_error_dummy_predict = mean_squared_error(y_test, y_test_dummy_pred)
mean_abs_error_dummy_predict = mean_absolute_error(y_test, y_test_dummy_pred)
print('mean squared error error = ' + str(mean_squared_error_dummy_predict))
print('mean absolute error = ' + str(mean_abs_error_dummy_predict))
```

Ridge Regression and evaluation of metrics.

```
reg = Ridge(alpha=150)
reg.fit(X_train, y_train)
y_test_ridge_pred = reg.predict(X_test)
mean_squared_error_ridge_predict = mean_squared_error(y_test, y_test_ridge_pred)
mean_abs_error_ridge_predict = mean_absolute_error(y_test, y_test_ridge_pred)
print('mean squared error error = ' + str(mean_squared_error_ridge_predict))
print('mean absolute error = ' + str(mean_abs_error_ridge_predict))
```

Random Forest Classifier and evaluation of metrics.

```
forest = RandomForestRegressor(n_estimators = 80, max_depth = 10)
forest.fit(X_train, y_train)
y_test_forest_pred = forest.predict(X_test)
mean_squared_error_forest_predict = mean_squared_error(y_test, y_test_forest_pred)
mean_abs_error_forest_predict = mean_absolute_error(y_test, y_test_forest_pred)
print('mean squared error error = ' + str(mean_squared_error_forest_predict))
print('mean absolute error = ' + str(mean_abs_error_forest_predict))
```