חיזוי מידת שיפור תוצאות ריצת המרתון בהסתמך על תוצאות מרוצי ההכנה

יעקב קמחי - 038820676

האוניברסיטה הפתוחה

סמינר גילוי ידע באמצעות כריית נתונים – 13802

מנחה: פרופ' רועי גלברד   
 תשפ"ג ,2023 סמסטר ב'

תוכן עניינים:

[שאלת המחקר 3](#_Toc141723756)

[כללי 3](#_Toc141723757)

[קצב הריצה בריצת המרתון 3](#_Toc141723758)

[סוגי הרצים מושאי המחקר 3](#_Toc141723759)

[סוגי מירוצים 3](#_Toc141723760)

[מרתונים בישראל 4](#_Toc141723761)

[מיפוי: נתונים-מידע-ידע 5](#_Toc141723762)

[סקר ספרות 5](#_Toc141723763)

[סקירת ספרות אודות עולם התוכן הנידון. גורמים משפיעים ומודלים מקובלים 5](#_Toc141723764)

[סקירת ספרות אודות כריית נתונים בעולם התוכן הנידון 6](#_Toc141723765)

[מקורות להגדרת טווח הערכים המקובל במדדים, לאורם תיבחן שאלת המחקר. 7](#_Toc141723766)

[שיטת המחקר 8](#_Toc141723767)

[פירוט המשתנים 8](#_Toc141723768)

[תוצאות 13](#_Toc141723769)

[פלט מפורט של ההרצה הטובה ביותר – זו שלגביה בוצעה בדיקת היציבות של המודל. 14](#_Toc141723770)

[פלט הרצת המודל כעץ החלטה j48 14](#_Toc141723771)

[דיון בתוצאות 16](#_Toc141723772)

[שימושים אפשריים במודול הידע שהתקבל 16](#_Toc141723773)

[מגבלות העבודה 16](#_Toc141723774)

[References 17](#_Toc141723775)

[נספחים 17](#_Toc141723776)

## שאלת המחקר

### כללי

הריצה למרחק המרתון נחשבת לריצה מאתגרת מאד, המצריכה הכנה מדוקדקת הכוללת הכנה נפשית, מטבולית, פיזיולוגית ואנטומית. חלק מן ההכנה נעשה כדיאטות ומניפולציות תזונתיות לאורך תקופת ההכנה, עיסויים, ניסויי ציוד וכו' אך החלק המרכזי בהכנה הינה ההכנה הפיזיולוגית והנפשית הנעשית באמצעות תוכנית האימונים. תוכנית האימונים, ככלל, מכילה העלאה הדרגתית בעומסי האימונים המגיעה לשיא, מספר שבועות לפני המרתון עצמו.

חלק בלתי נפרד בתוכנית ההכנה המקובלת לריצת המרתון, הינו מירוצי ההכנה המשולבים לאורך תוכנית האימונים. מירוצי ההכנה הינם (לרוב) מירוצים למרחקים הקטנים ממרחק המרתון (42195 מטרים) ובהם הרץ מנסה את כוחו ומתרגל מגוון פרוצדורות שאותם יבצע גם בריצת המרתון.

כרץ מרתון חובב וכמאמן ריצה (מוסמך ווינגייט), אני מתחבט בשאלת שיבוץ מירוצי הכנה בתוכנית האימונים למרתון. כגון: מהו מרחק המירוץ המומלץ ביותר (לדוגמא, ריצה ל-5 ק"מ או ריצה למרחק של 21 ק"מ), מהו אינטרוול הזמנים המוצלח ביותר בין מרוץ למרוץ ובין המרוץ למרתון עצמו, מהי מידת הקשר בין ההשקעה והשיפור במירוצי ההכנה לבין השיפור בתוצאות המרתון עצמו

על שאלות אלו ננסה לענות באמצעות הפקת מידע מקובץ נתוני המירוצים שנערכו בישראל בשנים 2016 עד 2018 ושנתקבל ממערכת מידע המאגדת בתוכה את כל תוצאות המירוצים שנערכים בישראל ומנגישה אותם לקהל באמצעות אתר האינטרנט [מבט למרוץ (raceview.net)](https://raceview.net/)

### קצב הריצה בריצת המרתון

שאלת קצב הריצה (כלומר, מהירות ליחידת זמן) הינה שאלה הרת גורל להצלחת\כישלון ריצת המרתון. באם קצב הריצה הינו מהיר מדי, לא רק שהרץ ייאלץ להפחית את קצב הריצה בשלבים המתקדמים של התחרות, אלא שלקו הסיום הוא יגיע בפיגור זמן משמעותי מכפי שהיה מגיע אילו היה רץ בקצב המדוייק שמתאים ליכולותיו. לעומת זאת באם ירוץ בקצב איטי מיכולותיו, כמובן שהוא יחצה את קו הסיום בתוצאה שהיא פחות טובה מכפי שיכל להשיג. מסיבה זו, תכנון וקביעת הקצב המדוייק לאותו הרץ, בתוואי הריצה ובאילוצי מזג האויר וכו', הינם שאלות שמעסיקות רצים ומאמנים השואפים למקסום יכולות הרץ.

### סוגי הרצים מושאי המחקר

הרצים בהם ארצה להתמקד, הינם רצים חובבנים (רצים שאינם ספורטאים המשתכרים מהשתתפות במירוצים) המעוניינים בשיפור זמני הריצה ובהגעה לתוצאה אישית המיטבית עבורם. הנחת היסוד הינה שריצת מרתון הינה תובענית המחייבת הכנה של מספר חודשים ולפיכך זמן ההכנה שנקבע כ"מרוץ הכנה" יהיו כאלו שנערכו במסגרת 120 הימים שטרם ריצת המרתון.

### סוגי מירוצים

ככלל, ישנם שלשה סוגי מירוצים עיקריים הנחלקים בעיקר לפי משטח הריצה אך מקפלים בתוכם סוגי הכנה ומרכיבי כושר גופני שונים בתכלית.

1. מירוצי כביש
2. מירוצי שטח.
3. מירוצי מסלול.

בחלוקה גסה ניתן לומר שמירוצי המסלול הינם לרצי עלית (שאינם נשואי מחקר זה כאמור)

מירוצי השטח גם הם לא נמדדים במחקר זה, הן מפני שמדידת המרחקים אינה עקבית לגביהם והן מפני ששטח הריצה הינו שונה ממרוץ למרוץ ואין יסוד להשוואה.

מאחר ומרכיב הכושר הגופני הפעיל ביותר והנדרש להצלחה בריצת המרתון הינו הסבולת האירובית, המירוצים שבהם יתמקדו רוב רצי ההישג הינם המירוצים שבהם נדרשת הסבולת האירובית ואלו הם המרוצים למרחקים ארוכים (מרחקים הגדולים מ-3000 מטרים). עם זאת, מרוץ הארוך ממרתון אינו יכול להחשב כמרוץ הכנה למרתון ולכן מגוון המרחקים שאותם נקח כפרמטר לחיזוי יהיו מרחקי מרוץ כביש הגדולים מ-3000 מטרים והקטנים או שווים למרחק המרתון (42195 מטרים)

### מרתונים בישראל

אירועי המרתון בישראל אינם רבים. האירועים שמופיעים במאגר הנתונים לשנים 2016-2018 הינם:

* מרתון ירושלים: אירוע מרתון עם כ-800 מטרים מצטברים ורטיקליים ההופכים את המרוץ לקשה באופן יחסי ליתר המרתונים.
* מרתון טבריה: אירוע המרתון הוותיק בישראל ומושך את מירב הרצים החובבנים באוריינטציה הישגית מאחר ומסלול הריצה הינו הישר והשטוח ביותר מבין יתר האירועים ולפיכך הוא מאפשר השגת תוצאות טובות יותר.
* מרתון תל אביב: מרתון שטוח באופן יחסי לירושלים אך עמוס יותר ובעל פניות רבות יותר.
* מרתון המדברי: מרתון הנערך באילת ויורד לשטח. מסלול הריצה אינו כביש אך שטוח למדי ומושך אליו רצי כביש.

## מיפוי: נתונים-מידע-ידע

טבלה 1 : מיפוי נתונים-מידע-ידע

|  |  |  |  |
| --- | --- | --- | --- |
| # | מאפיין\תהליך | בעולם האנושי – ארגוני | במערכת הממוכנת |
| 1 | נתונים | נתוני הרצים השונים וביצועיהם במרוצי הכביש | הפקת פלט נתוני כלל הרצים במרוצים השונים שנערכו בטווח השנים 2016-2018 |
| 2 | מידע | מיקום הרץ ביחס לאחרים באותו המרוץ | הפקת המידע בניתוח של ביצועי הרצים במרוץ מסויים או מרוצים שונים שרץ רץ מסויים. ניתוח מיקום הרץ ביחס לבני גילו, למגדר שאליו הוא משתייך וכו' |
| 3 | ידע | הסקת קצבי ריצה אפשריים במרוצים אחרים ע"ב נסיון. | מה תהיה מידת השיפור של אותו הרץ בריצת המרתון בהסתמך על נתוני ריצה קודמים של אותו הרץ. |
| 4 | הפיכת נתונים למידע וידע | הערכה על בסיס נסיון קודם, תחושה, צפיה באימוני הרץ. | בחירת מרחקי ריצה, בנקודת הזמן הנכונה על ציר תוכנית האימונים, וחיזוי תוצאת המרתון ע"ב תוצאת הריצה שהשיג הרץ במירוצי ההכנה. |
| 5 | הפיכת מידע וידע לנתונים | לא מתקיים להבנתי | לא מתקיים להבנתי |
| 6 | הפיכת ידע סמוי לגלוי | בקרת מאמן על תוכנית האימונים וביצוע שינויים בהתאם לביצועים במירוצי ההכנה | קבלת משוב בהתבסס על הזנת תוצאות מירוצי ההכנה. |
| 7 | הפיכת ידע גלוי לסמוי | לא ניתן | טעות נתונים שיכולה לקרות כתוצאה מהזנה שגויה. הוספת והפחתת משתנים |
| 8 | ידע כתובנה לתמיכה בהחלטות | קביעת קצב ריצה בריצת המרתון בהתבסס על תחושה | קביעת קצב הריצה בריצת המרתון בהתבסס על ידע שנרכש באמצעות אלגוריתם למידת המכונה. |
| 9 | ידע כתובנה התורמת לחידוש ולחדשנות | קביעת מירוצים שיבצעו הכנה מיטבית שתדמה את ריצת המרתון ואת תנאי המרוץ בדיוק הגבוה הניתן | קביעת מירוצים שיבצעו הכנה במרווח הזמן האופטימלי, למרחק הזמן האופטימלי ותוואי שטח האופטימלי בהתאם לממצאי המחקר. |
| 10 | למידה, חלוקה ושיתוף בידע | שיתוף של מאמנים ומתאמנים בפורומים ובמפגשים וירטואליים ופיזיים. | שיתוף באמצעות מערכת נגישה שתאפשר למידה ותחקור ע"ב נתוני ביצועים אפשריים ו\או שבוצעו. |

## סקר ספרות

### סקירת ספרות אודות עולם התוכן הנידון. גורמים משפיעים ומודלים מקובלים

כריית נתונים היא תהליך של גילוי תבניות ומגמות בנתונים, שנועדו על מנת לזהות דפוסים שאינם נראים לעין באופן מיידי. זה יכול להיות שימושי לעסקים, ממשלות וארגונים אחרים כדי לקבל החלטות טובות יותר, לשפר את השירות ללקוחות ולחזות את העתיד.

כריית נתונים מתבצעת על ידי שימוש בשיטות סטטיסטיות, מתמטיות ומכונאות למידה. הנתונים יכולים להיות ממקור אחד מסויים או ממקורות שונים. הנתונים מעובדים על מנת לזהות דפוסים, מגמות והקשרים בין המשתנים השונים.

ישנן מספר טכניקות שונות של כריית נתונים:

* ניתוח אשכולות: חלוקה של נתונים לקבוצות על סמך דמיון בין קבוצות.
* רגרסיה ליניארית: חיזוי ערך מטרה מספרי על סמך משתנים אחרים.
* ניתוח התאמה: זיהוי דפוסים בנתונים שאינם ידועים מראש.
* ניתוח מידע תיאורי: תיאור הנתונים בצורה שמקלה על ההבנה.

כריית נתונים היא תחום המתפתח במהירות, עם טכניקות חדשות שמפותחות כל הזמן. זהו כלי רב עוצמה שיכול לשמש לשיפור החיים בכל תחומי החיים.

כמה דוגמאות לשימושים של כריית נתונים:

* בעסקים: כריית נתונים יכולה לשמש לקבלת החלטות עסקיות טובות יותר, כגון:
  + זיהוי לקוחות פוטנציאליים
  + תמחור מוצרים ושירותים
  + שיווק ממוקד
  + ניהול מלאי
  + שירות לקוחות
* בממשלה: כריית נתונים יכולה לשמש לשיפור השירותים, כגון:
  + זיהוי פשעים וטרור
  + ניהול משאבים
  + קבלת החלטות פוליטיות
  + שירות לציבור
* בארגונים אחרים: כריית נתונים יכולה לשמש לשיפור השירותים, כגון:
  + בתי חולים
  + בתי ספר
  + מוזיאונים
  + מועדוני ספורט

חשוב לזכור שכריית נתונים אינה כלי מושלם. ישנם מספר אתגרים שצריך להתמודד איתם בעת השימוש בכריית נתונים, כגון:

* הטיה: כריית נתונים יכולה להיות מוטה על ידי הנתונים שבהם היא משתמשת. לדוגמה, אם נתוני האימון כוללים רק גברים, המודלים שהם מייצרים עשויים להיות מוטים לטובת גברים.
* פרטיות: כריית נתונים יכולה לשמש לאיסוף מידע אישי על אנשים. חשוב לוודא שהנתונים נאספים בצורה אתית ומכבדת את הפרטיות של האנשים.
* השפעה על ההתנהגות: כריית נתונים יכולה לשמש להשפיע על ההתנהגות של אנשים. לדוגמה, אם מודעה מותאמת אישית מוצגת לאדם, הוא עשוי לעשות את הפעולה שהמודעה מבקשת. על כן חשוב להשתמש בכריית נתונים בצורה אחראית ולא להשתמש בה כדי להשפיע על ההתנהגות של אנשים בצורה שלילית.

### סקירת ספרות אודות כריית נתונים בעולם התוכן הנידון

כפי שנכתב לעייל, בתחרות מרתון, אחד הגורמים החשובים ביותר להצלחה הוא קביעת קצב הריצה הנכון. קצב ריצה נכון יאפשר לרוץ את המרתון במהירות המיטבית תוך שמירה על מאגרי אנרגיה שיידרשו בשלבים המתקדמים של המרוץ.

כריית נתונים יכולה לשמש לקביעת קצב ריצה נכון על ידי ניתוח נתונים אישיים של הרץ, כגון:

* הזמן הטוב ביותר למרחקים קצרים יותר, כגון 5 ק"מ או 10 ק"מ
* הזמן שהרץ השיג בעבר במרתונים קודמים
* מרחק הזמן בין מירוצים (זמן התאוששות)

ניתוח הנתונים הללו יכול לשמש כדי ליצור מודל לחיזוי מהירות הריצה המירבית של הרץ במהלך המרתון. המודל יכול לשמש כדי לקבוע קצב ריצה מדויק שיאפשר לרץ להגיע ליעדו, שהוא לסיים את המרתון במהירות המיטבית תוך שמירה על כוחות.

על אף שכריית נתונים היא כלי שיכול לשמש לקביעת קצב ריצה הנכון ביותר בתחרות המרתון, חשוב לזכור שישנם מספר גורמים שיכולים להשפיע על ביצועי הרץ במרתון, שאינם נלקחים בחשבון במודל, כגון:

* מצב הרוח של הרץ ביום המרתון
* מצבו הבריאותי של הרץ ביום המרתון
* תנאי מזג האוויר ביום המרתון
* הפרשי משקל של הרץ בין הביצועים הקודמים לבין יום המרתון.

על הרץ והמאמן לקחת בחשבון את הגורמים הללו בעת קבלת ההחלטה על קצב הריצה במרתון. עם זאת, כריית נתונים יכולה לספק לרץ מידע חשוב שיכול לעזור לו לקבוע קצב ריצה נכון שיגדיל את סיכויי ההצלחה שלו במרתון.

#### מחקרים שנעשו בנושא

ישנם מספר מחקרים שנעשו בנושא שימוש בכריית נתונים לקביעת קצב ריצה בתחרות המרתון. מחקר אחד שפורסם בכתב העת "Sports Medicine" בשנת 2017 מצא שהמודלים שהשתמשו בכריית נתונים היו מסוגלים לחזות את זמן המרתון של הרצים עם דיוק של עד 95%. מחקר נוסף שפורסם בכתב העת "Medicine & Science in Sports & Exercise" בשנת 2018 מצא שהמודלים שהשתמשו בכריית נתונים היו מסוגלים לחזות את סיכויי סיום מרתון של הרצים עם דיוק של עד 85%. מחקר נוסף שפורסם בכתב העת "Journal of Strength and Conditioning Research" בשנת 2019 מצא שהמודלים שהשתמשו בכריית נתונים היו מסוגלים לחזות את השיפור בזמן המרתון של הרצים עם דיוק של עד 75%.

המחקרים הללו מראים כי השימוש בכריית נתונים יכול להיות יעיל בקביעת קצב ריצה בתחרות המרתון. עם זאת, חשוב לציין שהמודלים שהשתמשו במחקרים הללו לא היו מושלמים, וישנם מספר גורמים שיכולים להשפיע על ביצועי הרץ במרתון, שאינם נלקחים בחשבון במודלים הללו.

### מקורות להגדרת טווח הערכים המקובל במדדים, לאורם תיבחן שאלת המחקר.

#### משוואת הרגרסיה

forecast (predicted) = AX1 + BX2 + CX3 + DX4 + EX5 + ... + F

הנוסחה הזו היא מודל רגרסיה ליניארי שמשתמש בנתונים היסטוריים כדי לנבא מכירות עתידיות. הנוסחה מחלקת את הנתונים לשני סוגים: משתנים עצמאיים ומשתנים תלויים. המשתנים העצמאיים הם הנתונים שיש לנו, והמשתנים התלויים הם הנתונים שאנחנו מנסים לנבא. במקרה הזה, המשתנים העצמאיים הם מרחקי המירוצים שבוצעו במהלך ההכנה, מרווחי הזמן שבין המירוצים ומירוץ היעד A, B, C, D, E ו-F. המשתנה התלוי הוא תחזית זמן הריצה.

צורת עבודת הנוסחה:

1. היא מחשבת את הערך של כל הפרמטרים A, B, C, D, E ו-F.
2. היא משתמשת בערכים של הפרמטרים כדי לחשב את התחזית.

התחזית היא הערך המשוער של המשתנה התלוי. במקרה הזה, התחזית היא הערך המשוער של השיפור החזוי בריצת המרתון.

#### מדד הערכה

מדד הערכה למודל הוא שורש ממוצע ריבועי הטעויות - Root mean squared error RMSE  
שזה מדד של המרחק הממוצע בין נקודות הנתונים למערך התחזית.

במקרה של חיזוי ריצת המרתון, המודל ינסה לחזות את זמן ריצת המרתון של הרץ. ה-RMSE יהיה המרחק הממוצע בין זמן המרתון המשוער לזמן המרתון בפועל. ערך RMSE נמוך פירושו שהמודל מדויק יותר בחיזוי זמן המרתון של הרץ.

## שיטת המחקר

### פירוט המשתנים

כאמור, מקור הנתונים שעליו התבסס מחקר זה, הינו נתוני ביצועי הרצים במירוצים השונים שנערכו בישראל בשלש השנים – 2016-2018. המירוצים הינם למרחקי ריצה שונים – החל מ100 מטרים וכלה במרחקי אולטרה-מרתון שנערכים למרחקים של 200 ק"מ ואפילו יותר מכך.

נעשו פעולות הכנה ועיבוד מידע מקיפות שבסופן התקבל מאגר מידע מטוייב המכיל נתוני מירוצי כביש במקצי מרחקי הריצה: 5km, 10km, 15km, 21.1km, 42.2km

חילוץ הנתונים נעשה באמצעות אלגוריתם שביצע HTTP Requests. המשך עיבוד הנתונים נעשה בקוד שנכתב בשפת Python וכן בגליונות אקסל.

#### תיאור הנתונים, מקורם, ושימושים קודמים (אם שימשו במחקרים אחרים)

המשתנים שחושבו בהתבסס על הנתונים הגולמיים ואשר נבחרו בסופו של דבר לשמש את ריצת האלגוריתם, הינם:

#### Marathon

אירוע המרתון שבו הושגו תוצאות מאגר המידע. ערך נומינלי המכיל את מגוון אירועי המרתון שפורטו בחלק הראשון.

#### RaceOneKM

מרחק המרוץ המאוחר מבין מירוצי ההכנה. ערך נומינלי המכיל את סדרת הערכים: (5000, 10000, 15000, 21000, 21095, 42195)

#### TimeBetweenMarathonAndRaceOne

אינטרוול הזמן שבין אירוע המטרה (המרתון) ובין מרוץ ההכנה המאוחר ביותר שלפניו (הערך הינו ערך מספרי שלילי ולפיכך בפונקציית הרגרגסיה נצפה לראות אותו עם מקדם שלילי)

#### RaceTwoKM

מרחק המרוץ המוקדם מבין שני מירוצי ההכנה. ערך נומינלי המכיל את סדרת הערכים: (5000, 10000, 15000, 21000, 21095, 42195)

#### TimeBetweenRaceOneAndRaceTwo

אינטרוול הזמן שבין המרוץ המאוחר מבין שני מרוצי ההכנה למרתון ובין מרוץ ההכנה שלפניו (הערך הינו ערך מספרי שלילי ולפיכך בפונקציית הרגרגסיה נצפה לראות אותו עם מקדם שלילי)

#### ImprovementRate

ערך המדידה של השיפור שהושג. הערך הינו ערך מחושב של ערך המנורמל של תוצאת המרתון ביחס לתוצאות כלל הרצים לאותו המרוץ (ע"פ מספר שנים) בהפחתת ממוצע התוצאות המנורמלות של מרוצי ההכנה. כלומר, אם הערך הינו קטן מ-0, בוצע שיפור בביצועי הרץ במרוץ המטרה לעומת הביצועים במרוצי ההכנה. נוסחת החישוב:

#### פירוט המניפולציות שהופעלו בייצוג הנתונים (סינון, נירמול, קטגוריזציה)

הנתונים שהתקבלו מהאתר היו כדלהלן:

טבלה 2: פירוט המניפולציות וייצוג הנתונים

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| שם השדה | תיאור השדה | ערך לדוגמה | | מניפולציה שבוצעה | | שם שדה סופי | | תיאור השדה | ערך לדוגמה |
| מירוץ הנהנתנים | אירוע הריצה | מירוץ הנהנתנים | | מחיקת רווחים + קטגוריזציה | | Marathon | | שם מרוץ המרתון | JerusalemMarathon |
| |  |  | | --- | --- | | RaceOne | RaceOne | | | שם המרוץ המאוחר יותר מבין מרוצי ההכנה | מירוץהנהנתנים |
| RaceTwo | | שם המרוץ המוקדם יותר מבין מרוצי ההכנה | מירוץסובבתבור |
| Event\_date | תאריך האירוע | 12/03/2016 | |  | | MarathonDate | | תאריך האירוע במדה וזהו המרתון | 12/03/2016 |
| Dist | מרחק המקצה | | 5K | | Numerization (הפיכת השדה למספרי המוצג במטרים. לדוגמא 5000) | RaceOneKM | מספר הק"מ במדה וזהו המרוץ המוקדם מבין מרוצי ההכנה | | 21095 | |
| RaceTwoKM | מספר הק"מ במדה וזהו המרוץ המאוחר יותר מבין מרוצי ההכנה | | 10000 | |
| first\_name |  | יקי | | Concatenation of the names with the age and Gender in order to creating Run nerIdentigfier | | RunnerID | | מזהה רץ ח"ע | יקיקמחי1983M |
| Last\_name |  | קמחי | |
| Results |  | 0:25:21 | | MarathonResult | | תוצאה במדה וזהו מרחק המרתון | | פורמט המקור – hh:mi:sec | 3:52:29 |
| RaceOneResult | | תוצאה במדה וזהו מרוץ ההכנה המאוחר יותר | | פורמט המקור – hh:mi:sec | 1:10:58 |
| RaceTwoResult | | תוצאה במדה וזהו מרוץ ההכנה המוקדם יותר | | פורמט המקור – hh:mi:sec | 00:40:58 |
| RaceOneResultInSeconds | | תוצאת המרוץ המוקדם בפורמט שניות | | In Seconds | 5151 |
| RaceTwoResultInSeconds | | תוצאת המרוץ המאוחר בפורמט שניות | | In seconds | 7575 |
| MarathonResultInSeconds | | תוצאת המרתון בפורמט שניות | | In seconds | 7777 |
| MarathonResultInSecondsNormalized | | תוצאת המרתון בנרמול הנתונים | | Normalize | 0.314286 |
| RaceOneResultInSecondsNormalized | | תוצאת המרוץ המוקדם בנירמול הנתונים | | Normalize | 0.554286 |
| RaceTwoResultInSecondsNormalized | | תוצאת המרוץ המוקדם בנירמול הנתונים | | Normalize | 0.3444 |
| ImprovementRate | | חישוב השדה המנורמל של המרתון בהפחתת ממוצע התוצאות המנורמלות האחרות | | Normalize + calculation | -0.01 |
| Privet\_result |  | NULL | |  | |  | |  |  |
| Sex |  | F | |  | |  | |  |  |
| Birth\_Date |  | 12/03/1983 | |  | |  | |  |  |
| Raceid |  | 27844 | |  | |  | |  |  |
| Eventid |  | 8246 | |  | |  | |  |  |

#### מדדי ההערכה

מדד הערכה שלפיו נעריך את השיפור הצפוי הוא שורש ממוצע ריבועי הטעויות - Root mean squared error RMSE של הערך הנחזה – שיפור הרץ בעקבות נתוני מרוצי ההכנה.   
RMSE זהו מדד של המרחק הממוצע בין נקודות הנתונים למערך התחזית כאשר התחזית הינה ערך המדידה של השיפור שהושג. הערך הינו ערך מחושב של ערך המנורמל של תוצאת המרתון ביחס לתוצאות כלל הרצים לאותו המרוץ (ע"פ מספר שנים) בהפחתת ממוצע התוצאות המנורמלות של מרוצי ההכנה. כלומר, אם הערך הינו קטן מ-0, בוצע שיפור בביצועי הרץ במרוץ המטרה לעומת הביצועים במרוצי ההכנה. נוסחת החישוב:

במקרה של חיזוי ריצת המרתון, המודל ינסה לחזות את זמן ריצת המרתון של הרץ. ה-RMSE יהיה המרחק הממוצע בין זמן המרתון המשוער לזמן המרתון בפועל. ככל והמספר יהיה קטן, כך החיזוי הינו מדוייק יותר.

#### האלגוריתמים והערכים שהוזנו בפרמטרים השונים שלהם

בבואי להשתמש בתוכנת Weka להרצת המודלים ומדידתם, ראשית ביצעתי הסרה של נתונים בהם לא ניתן לזהות את הרץ (כגון ששנת הלידה שלו אינה מופיעה). או שהנתונים אינם מציאותיים (תוצאת מרתון בפחות משעתיים) ועוד פעולות רבות של ניקוי הנתונים. הכלים שסייעו בביצוע ההכנה, הניקוי ומיטוב הנתונים הינם פרוצדורות שנכתבו בשפת Python ומאוחר יותר אף שימוש בגליונות Excel

שלב שני, בעצת המנחה (פרופ' רועי גלברד) היה להעביר את התוצאות מרמת השורות לרמת העמודות. כך לדוגמא

פורמט המקורי:

טבלה 3: דוגמא למבנה הנתונים במקור

|  |  |  |  |
| --- | --- | --- | --- |
| רץ | מרחק | תאריך | תוצאה בשניות |
| 1234 | 42195 | 01.01.2018 | 5555 |
| 1234 | 10000 | 01.12.2017 | 1000 |
| 1234 | 5000 | 01.11.2017 | 400 |
|  |  |  |  |

פורמט היעד

טבלה 4: דוגמא למבנה הנתונים המעובד לשם ריצת התוכנית

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| רץ | תוצאת המרתון | תוצאה מנורמלת | תאריך המרתון | מרחק מרוץ ראשון | הפרש ימים ממרוץ שני | תוצאת מרוץ ראשון | מרחק מרוץ שני | הפרש ימים מהמרתון | תוצאת מרוץ שני |
| 1234 | 5555 | 0.12 | 01.01.2018 | 5000 | 30- | 400 | 10000 | -30 | 1000 |

חישוב נוסף שנעשה הינו השיפור שעשה הרץ – ImprovemenRate שהוא בוצע באמצעות הפחתת ממוצע התוצאות המנורמלות של שני המרוצים המקדימים מהתוצאה המנורמלת של המרתון כך שהתקבלה הפונקציה שתוארה לעייל ושבאמצעותה נעריך את הצלחתו או אי הצלחתו של הרץ בשיפור זמני הריצה ביחס למתחרים האחרים.

לבסוף היה נדרש לייצר קובץ Arff כך שהערכים יהיו נומינליים וניתן יהיה לבצע ניתוח השפעת הערכים הנומינליים (מרחק המרוץ בק"מ, שם המרוץ וכו') הדבר היה כרוך בעבודה ידנית ומאומצת לאור שגיאות רבות בהמרת הקובץ מפורמט CSV לפורמט ARFF.

##### בחירת המודל

נבדקו מספר מודלים – מודל עצי החלטה J48 ולעומתו מודל Linear Regression.

לצורך השימוש במודל עצי ההחלטה, בוצע חלוקה של משתנה היעד ImprovementRate לשתי אפשרויות – אם הוא קטן מ0 (כלומר המיקום היחסי של הרץ במרתון הינו גבוה מממוצע מיקומו היחסי במרוצי ההכנה, אזי המשתנה Improvement אוכלס בערך Success, אך אם הוא גדול מ-0 המשתנה אוכלס בערך Failure.

בהתבוננות בויזואליזציה של העץ ובהרצה תחת משתנים שונים, היה ברור כי עץ החלטה אינו מספק ערך ריאלי.

לעומת זאת בהרצות שבוצעו במודל שנבחר - רגרסיה לניארית התוצאות – הן של ה- RMSE והן של המודל היו משביעות רצון.

##### הרצות המודל

מתוך האפשרויות להגדרת בחירת המשתנים בתוכנה לצורך הריצה NoAttributeSelection:attributeSelectionMethod - קבע את השיטה כדי לבחור תכונות לשימוש רגרסיה ליניארית. בחרתי להשתמש: ללא בחירת משתנים

. False:Debug - ללא ניפוי טעויות ופלטי מידע.

False:eliminateColinearAttributes- ללא השמטה של משתנים לא ליניאריים

#### שיטת "הרעשת" הנתונים לצורך בדיקת רגישות- יציבות המודל.

על מנת לבחון את רגישות - יציבות המודל – בוצעה הוספת ערכים למודל החיזוי על ידי הגדלת בגבולות של בין 0 % ועד 30 % למשתנה ImprovementRate שאותו אנו מנסים לחזות.

ההרעשה בוצעה על ההרצה האופטימלית שאותה בחרנו כמודל הנבחר.

בכדי לבצע את ההרעשה השתמשנו בגיליון אלקטרוני .

להלן התוצאות בערכים של RMSE:

טבלה 5: פירוט תהליך ה"הרעשה"

|  |  |
| --- | --- |
| אחוז ההגדלה | ערך RMSE |
| 0% | 0.076 |
| 5% | 0.0798 |
| 10% | 0.0836 |
| 15% | 0.0874 |
| 20% | 0.0912 |
| 25% | 0.095 |
| 30% | 0.0988 |

על אף ההרעשה המודל הציג יציבות.

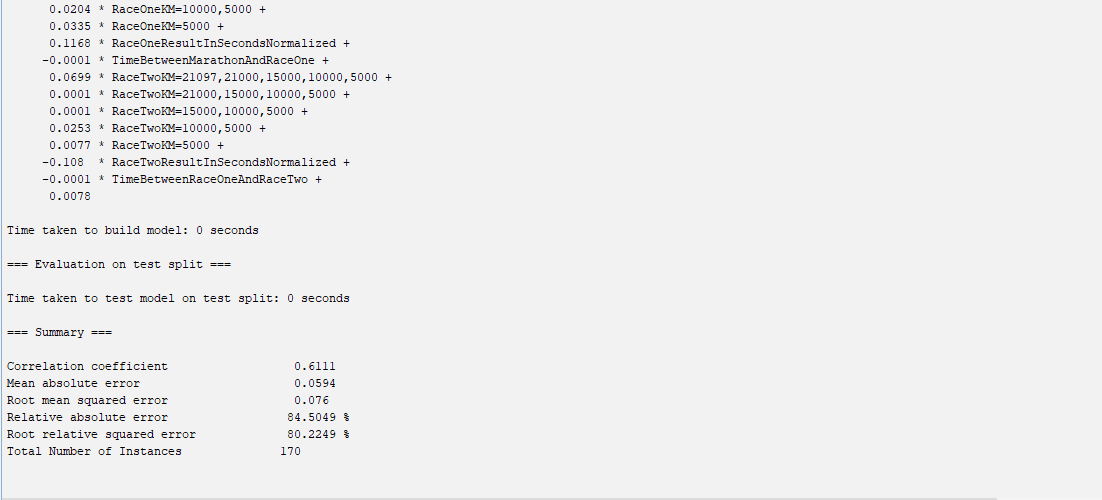
## תוצאות

טבלה 6: פירוט תוצאות הרצת המודלים

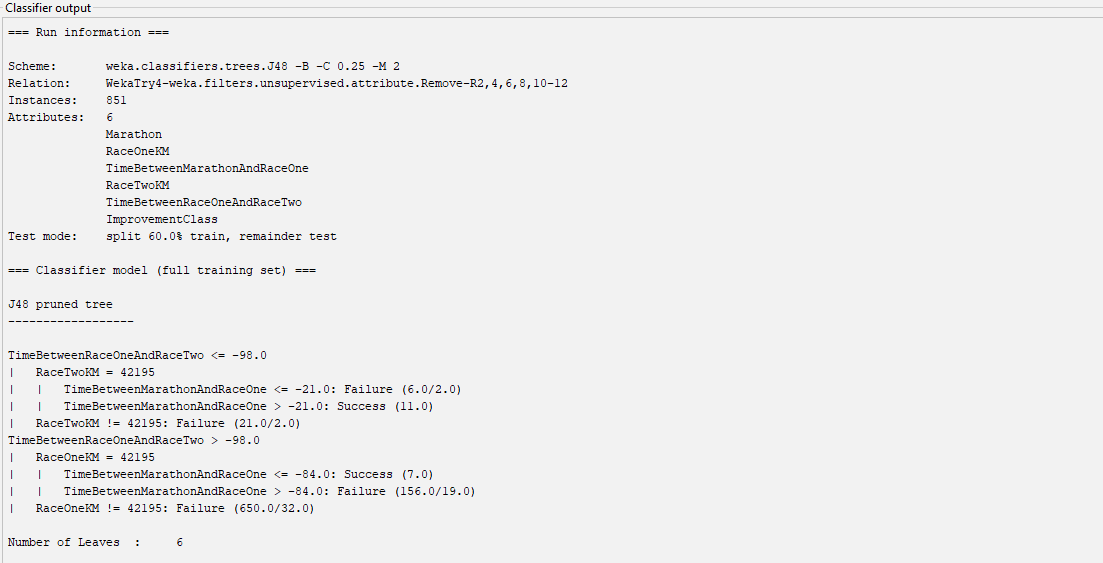
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | DataSet | Algorithm | F-score | RMSE |
| 1 | Marathon  RaceOneKM RaceOneResultInSecondsNormalized TimeBetweenMarathonAndRaceOne  RaceTwoKM RaceTwoResultInSecondsNormalized TimeBetweenRaceOneAndRaceTwo  ImprovementRate | Linear Regression | - | 0.76 |
| 2 | Marathon  RaceOneKM TimeBetweenMarathonAndRaceOne  RaceTwoKM TimeBetweenRaceOneAndRaceTwo  ImprovementClass | J48 | 0.926 | 0.235 |
| 3 | Marathon  RaceOneKM TimeBetweenMarathonAndRaceOne  RaceTwoKM TimeBetweenRaceOneAndRaceTwo  ImprovementRate | Linear Regression | - | 0.0762 |
| 4 | RaceOneKM RaceOneResultInSecondsNormalized TimeBetweenMarathonAndRaceOne  RaceTwo  RaceTwoKM RaceTwoResultInSecondsNormalized TimeBetweenRaceOneAndRaceTwo ImprovementRate | Linear Regression | - | 0.0854 |
| 5 | RaceOneKM TimeBetweenMarathonAndRaceOne  RaceTwo  RaceTwoKM TimeBetweenRaceOneAndRaceTwo  ImprovementRate | Linear Regression | - | 0.0851 |
| 6 | RaceOneResultInSecondsNormalized RaceTwoResultInSecondsNormalized  ImprovementRate | Linear Regression | - | 0.0932 |
| 7 | Marathon RaceOneResultInSecondsNormalized TimeBetweenMarathonAndRaceOne RaceTwoResultInSecondsNormalized TimeBetweenRaceOneAndRaceTwo  ImprovementRate | Linear Regression | - | 0.0865 |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |

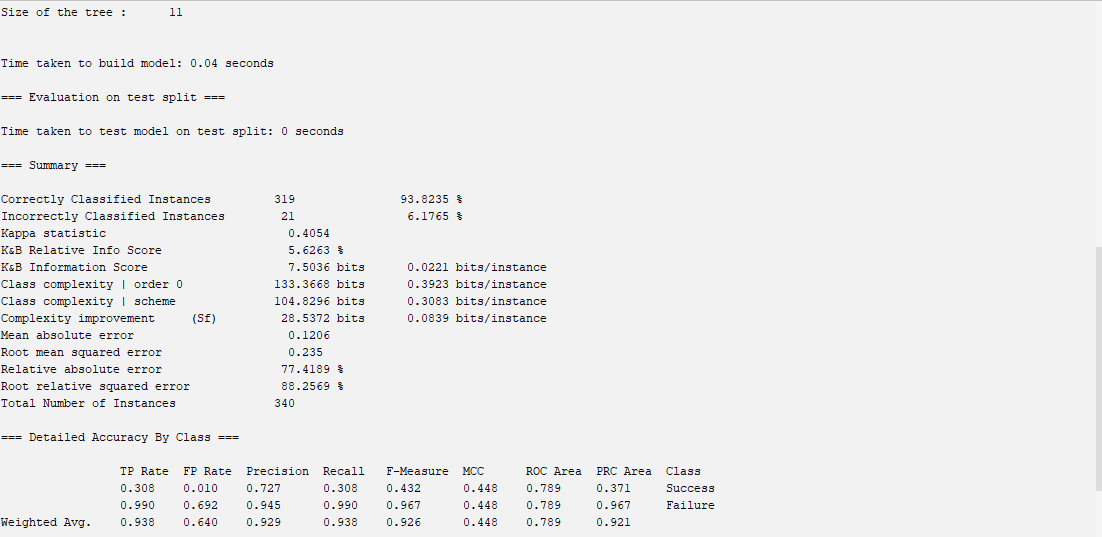
### פלט מפורט של ההרצה הטובה ביותר – זו שלגביה בוצעה בדיקת היציבות של המודל.





### פלט הרצת המודל כעץ החלטה j48





A diagram of a company

Description automatically generated

### דיון בתוצאות

אני בחרתי שלא להמשיך עם מודל העץ, על אף שהמודל הזה הראה ערכים טובים בסולם הRMSE, לאחר ניתוח עץ ההחלטה שהתקבל. העץ שהתקבל אינו "עושה שכל" מאחר וכל התלויות שהוצגו אינם עולים בקנה אחד עם הספרות וכן התוצאה שהייתי מצפה לקבל כרץ וכמאמן, אינה בינארית אלא משתנה רציף וכמותי המכמת את כמות השיפור שהושגה ע"י הרץ.

לעומת זאת, כאשר בחרתי במודל הרגרסיה הליניארית, והוספתי משתנים (כהרעשת המודל) שמהידע המקצועי שלי ידוע כי הקשר בינם לבין כמות השיפור אינה קיימת, אכן התוצאות שהתקבלו הינם עליה בערכי RMSE כמצופה.

### שימושים אפשריים במודול הידע שהתקבל

השימוש שמתוכנן להיעשות הינו העלאה של המודל והנגשתו לציבור הרצים, כך שיבחרו את הפרמטרים הבאים:

1. שם הרץ, שנת לידה, מגדר (מתוך רשימת בחירה)
2. מרתון היעד

התוכנית תבצע איסוף של מירוצי העבר של הרץ (בתוך טווח 120 הימים שקודמים למרתון היעד). במדה ואין מספיק נתונים, תופיע הודעת חיווי. במדה ויש נתונים במערכת עבור אותו הרץ, המערכת תבצע את החישובים המקדימים שבוצעו (כפי שניתן לראות במקטע ה"נספחים"), תריץ את המודל עבור המשתנים ותאסוף את הנתונים.

פלט התוכנית יהיה תוצאת מרתון חזויה ובכללה קצב הריצה.

### מגבלות העבודה

אין לנו מידע לגבי "כישלון" של רץ או באיזו מידה הוא השקיע ביום המרוץ, כך שהנתונים הינם הקפאת תמונת מצב שייתכן ואינה משקפת ולפיכך האופן שבו נדרש ליישם את המודל הוא לבקש מהמשתמש לדרג את מירוצי העבר שלו באמצעות סולם מאמצים מוערך שלו, ולקחת את הערכים שבהם המאמץ שהמשתמש דיווח שבהם הוא התאמץ באופן סאב-מקסימלי (כדי להעריך נכונה מצב אמת נתון).

האלגוריתם יאמן את עצמו ויוכל להשיא ערך לקהילת הרצים ולסייע להם במיקסום הביצועים ביום המרתון.

## References

Havenar, J., & Lochbaum, M. (2007). Differences in participation motives of first-time marathon finishers and pre-race dropouts. *Psychology of Sport and Exercise, 8(3), 297-316.*

This study examined the differences in participation motives between first-time marathon finishers and pre-race dropouts. The researchers found that finishers were more likely to be motivated by personal challenge, while dropouts were more likely to be motivated by social pressure.

Martin, D. T., Puetz, T., & Puetz, J. (2009). Predicting marathon performance from pre-race characteristics. *Journal of Strength and Conditioning Research, 23(1), 265-270.*

This study investigated the relationship between pre-race characteristics and marathon performance. The researchers found that factors such as age, sex, body mass index, and previous marathon experience were all significant predictors of marathon performance.

Seiler, S., & Tsolakidis, D. (2010). Predicting marathon performance from half-marathon time. *European Journal of Applied Physiology, 108(5), 809-816.*

This study examined the validity of using half-marathon time to predict marathon performance. The researchers found that half-marathon time was a significant predictor of marathon performance, with an average error of 2.1%.

Weir, J. P., & Noakes, T. D. (1998). Prediction of marathon performance from 10-km race time. *British Journal of Sports Medicine, 32(4), 240-244.*

This study investigated the relationship between 10-km race time and marathon performance. The researchers found that 10-km race time was a significant predictor of marathon performance, with an average error of 3.1%.

## נספחים

#### נספח 1: קוד פייתון של קובץ העריכה ועיבוד הנתונים הראשון

#### 

#### נספח2: קוד Python שנכתב לשם עיבוד הנתונים לכדי מבנה שיאפשר את הרצת האלגוריתם

import pandas as pd

from datetime import datetime, timedelta

import numpy as np

def process\_race\_data(input\_file\_path, output\_file\_path):

        # Read the Excel file

    df = pd.read\_excel(input\_file\_path)

    # Sort the data by RunnerID and DateOfTheRace

    df.sort\_values(by=['RunnerID', 'DateOfTheRace'], ascending=[True, True], inplace=True)

    # Create a new DataFrame to store the results

    output\_data = pd.DataFrame(columns=[ 'Marathon','RaceOne','RaceOneKM','RaceOneResultInSecondsNormalized',

                                        'TimeBetweenMarathonAndRaceOne','RaceTwo','RaceTwoKM',

                                        'RaceTwoResultInSecondsNormalized','TimeBetweenRaceOneAndRaceTwo',

                                        'MarathonResultInSeconds','MarathonResultInSecondsNormalized','ImprovementRate',

                                        'ImprovementClass',

                                        'RunnerID','MarathonDate', 'MarathonResult', 'RaceOneResult', 'RaceOneResultInSeconds',

                                        'RaceTwoResult', 'RaceTwoResultInSeconds'])

    # Iterate over unique RunnerIDs\

    max\_time\_per\_race = df.groupby(['NameOfTheRace', 'NumberOfMeters'])['Result'].max()

    min\_time\_per\_race = df.groupby(['NameOfTheRace', 'NumberOfMeters'])['Result'].min()

    for runner\_id in df['RunnerID'].unique():

        runner\_data = df[df['RunnerID'] == runner\_id]

        # Find the marathon (race with distance > 40000 meters)

        marathon\_data = runner\_data[runner\_data['NumberOfMeters'] > 40000].tail(1)

        if marathon\_data.empty:

            continue

        marathon\_name =  marathon\_data['NameOfTheRace'].values[0]

        marathon\_date =  marathon\_data['DateOfTheRace'].values[0]

        marathon\_result = marathon\_data['Result'].values[0]

        marathon\_result\_seconds = time\_to\_seconds(marathon\_result)

        marathon\_date = marathon\_data['DateOfTheRace'].values[0]

        # Find the races before the marathon

        races\_before\_marathon = runner\_data[runner\_data['DateOfTheRace'] < marathon\_date ]

        if len(races\_before\_marathon) >= 2:

            race\_one = races\_before\_marathon.tail(1)

            race\_two = races\_before\_marathon.tail(2).head(1)

            race\_one\_name = race\_one['NameOfTheRace'].values[0]

            race\_one\_result = race\_one['Result'].values[0]

            race\_one\_result\_seconds = time\_to\_seconds(race\_one\_result)

            race\_one\_distance = race\_one['NumberOfMeters'].values[0]

            race\_one\_date = race\_one['DateOfTheRace'].values[0]

            race\_two\_name = race\_two['NameOfTheRace'].values[0]

            race\_two\_result = race\_two['Result'].values[0]

            race\_two\_result\_seconds = time\_to\_seconds(race\_two\_result)

            race\_two\_distance = race\_two['NumberOfMeters'].values[0]

            race\_two\_date = race\_two['DateOfTheRace'].values[0]

            # Normalization calculation

            min\_race\_one = time\_to\_seconds(min\_time\_per\_race[(race\_one\_name,race\_one\_distance)])

            min\_race\_two = time\_to\_seconds(min\_time\_per\_race[(race\_two\_name,race\_two\_distance)])

            min\_marathon =  time\_to\_seconds(min\_time\_per\_race[(marathon\_name,42195)])

            range\_race\_one =  time\_to\_seconds(max\_time\_per\_race[(race\_one\_name,race\_one\_distance)]) - min\_race\_one

            range\_race\_two = time\_to\_seconds(max\_time\_per\_race[(race\_two\_name,race\_two\_distance)]) - min\_race\_two

            range\_marathon = time\_to\_seconds(max\_time\_per\_race[(marathon\_name,42195)]) - min\_marathon

            if range\_race\_one == 0 or range\_marathon == 0 or range\_race\_two == 0:

                # to not devide by zero

                continue

            race\_one\_normalized = (race\_one\_result\_seconds - min\_race\_one) / range\_race\_one

            race\_two\_normalized = (race\_two\_result\_seconds - min\_race\_two) / range\_race\_two

            marathon\_normalized = (marathon\_result\_seconds - min\_marathon) / range\_marathon

            time\_between\_marathon\_and\_race\_one = (race\_one\_date - marathon\_date).astype('timedelta64[D]')/np.timedelta64(1, 'D')

            time\_between\_race\_one\_and\_race\_two = (race\_two\_date - race\_one\_date).astype('timedelta64[D]')/np.timedelta64(1, 'D')

        else:

            # If there are no two races before the marathon, skip this runner

            continue

        if time\_between\_marathon\_and\_race\_one + time\_between\_race\_one\_and\_race\_two <= -120 :

            # if the time range between the marathon and race two is more than 4 months - it's not preperation for marathon

            continue

        output\_data = pd.concat([output\_data, pd.DataFrame([{

            'RaceOneResultInSecondsNormalized': race\_one\_normalized,

            'RaceTwoResultInSecondsNormalized': race\_two\_normalized,

            'MarathonResultInSecondsNormalized': marathon\_normalized,

            'RunnerID': runner\_id,

            'Marathon': marathon\_name,

            'MarathonDate': marathon\_date,

            'MarathonResult': marathon\_result,

            'MarathonResultInSeconds': marathon\_result\_seconds,

            'RaceOne': race\_one\_name,

            'RaceOneResult': race\_one\_result,

            'RaceOneKM': race\_one\_distance,

            'RaceOneResultInSeconds': race\_one\_result\_seconds,

            'TimeBetweenMarathonAndRaceOne': time\_between\_marathon\_and\_race\_one,

            'RaceTwo': race\_two\_name,

            'RaceTwoResult': race\_two\_result,

            'RaceTwoKM': race\_two\_distance,

            'RaceTwoResultInSeconds': race\_two\_result\_seconds,

            'TimeBetweenRaceOneAndRaceTwo': time\_between\_race\_one\_and\_race\_two

        }])], ignore\_index=True)

         # Calculate the maximum values for normalization

        #max\_marathon\_result = output\_data['MarathonResultInSeconds'].max()

        #max\_race\_one\_result = output\_data['RaceOneResultInSeconds'].max()

        #max\_race\_two\_result = output\_data['RaceTwoResultInSeconds'].max()

        # Calculate normalized values for MarathonResultInSeconds, RaceOneResultInSeconds, and RaceTwoResultInSeconds

        #output\_data['MarathonResultInSecondsNormalized'] = output\_data['MarathonResultInSeconds'] / max\_marathon\_result

        #output\_data['RaceOneResultInSecondsNormalized'] = output\_data['RaceOneResultInSeconds'] / max\_race\_one\_result

        #output\_data['RaceTwoResultInSecondsNormalized'] = output\_data['RaceTwoResultInSeconds'] / max\_race\_two\_result

    output\_data['ImprovementRate'] = output\_data['MarathonResultInSecondsNormalized'] - ( output\_data['RaceOneResultInSecondsNormalized'] + output\_data['RaceTwoResultInSecondsNormalized']) /2

    #output\_date['ImprovementClass'] = 'Success' if output\_data['ImprovementRate'] < 0 else 'Not Success'

    output\_data.to\_csv(output\_file\_path, index=False, encoding='utf-8-sig')

def time\_to\_seconds(time\_str):

    # Convert a time string in format 'hours:minutes:seconds' to total seconds

    time\_str = time\_str.strftime("%H:%M:%S")

    h, m, s = map(int, time\_str.split(':'))

    return h \* 3600 + m \* 60 + s

# Example usage:

input\_file\_path = 'InputToProcess.xlsx'

output\_file\_path = 'OutputFromProcess.csv'

process\_race\_data(input\_file\_path, output\_file\_path)

#### נספח 4: קובץ הARFF שעליו בוצעו המודלים



#### נספח 5: קבצי הרעשה לדוגמא



