

AI Project  
Fantasy Premier League Player  
פרויקט בבינה מלאכותית  
שחקן פנטזי פריימרליג

הפקולטה למדעי המחשב, טכניון  
אוקטובר 2021



ניסן אוחנה	ניב שפק	איתמר זיס
312332414	311141519	312238637
nissan.ohana@campus.technion.ac.il	Niv.shapk@campus.technion.ac.il	itamar.ziss@campus.technion.ac.il

**Github Repo**

# AI Project - Fantasy Premier League Player

## Contents

3	מבוא	1
4	רקע – פנטזי פריימרליג	2
8	הגדרת הבעיה	3
10	תיאור המערכת	4
11	הצגה כללית של המערכת וסימולטור המשחק	4.1
13	בסיס הנתונים - בנייתו ועיבוד מקדים	4.2
15	Expected Points Model – חיזוי נקודות ע"פ סטטיסטיקות ונתוני המשחק	4.3
15	4.3.1 גישה סטאטית	
16	4.3.2 גישה דינאמית	
16	4.3.3 תיאור המודל הנבחר	
19	Media Model – חיזוי כשירות ו"הייפ"	4.4
28	The Player Algorithm – מימוש השחקן והפעולות במשחק	4.5
32	אלגוריתמים ושיטות נוספות במערכת	4.6
	4.6.1 כתיב פונטי	32
34	ניסויים, כיוון הפרמטרים ותוצאות	5
34	הצגה כללית של תוכנית הניסויים	5.1
35	ניסויים וכיוון מודל חיזוי הנקודות	5.2
47	ניסויים וכיוון מודל המדיה	5.3
48	ניסוי 1: stop words	5.3.1
49	ניסוי 2: LSTM vs GRU	5.3.2
50	ניסוי 3: Dropout, Pooling	5.3.3
51	תוצאות, ניסויים וכיוון אלגוריתם השחקן	5.4
56	מסקנות	6
57	הצעות לשיפור, פיתוח עתידי וסיכום	7
59	נספח - מפרט טכני, מדריך למשתמש ולבודק	8

# AI Project - Fantasy Premier League Player

## 1. מבוא

פנטזי פריימרליג (FPL) הוא אחד ממשחקי הספורט-אסטרטגיה הפופולריים יותר בעולם. במשחק, עלינו להרכיב סגל שחקנים דמיוני מקבוצות הפריימרליג האנגלית, והסגל שהעמדנו צובר נקודות לפי הישגי השחקנים הפועל ותרומתם בכל מחזור משחק. האתגר על המשתמש הוא בבחירת ההרכב, אשר כולל הגבלות רבות ועליו לקבל החלטות אסטרטגיות בכל שבוע עבור דרך הפעולה שלו: חילופי שחקנים, הפעלת יכולות מיוחדות ועוד. המטרה, כמובן, היא לצבור את מירב הנקודות. מרחב האפשרויות עצום, ותלוי בגורמים רבים, מה שהופך את המשחק למעניין ומאתגר במיוחד. מטרתנו היא לבנות שחקן איכותי, המבוסס על מודל מאומן, אשר יבא את הפעולות שעליו לבצע בכל מחזור של משחק על מנת למקסם את הנקודות וההישגים במשחק.

מטרת הפרויקט היא לממש שחקן איכותי שיצבור כמה שיותר נקודות. בצענו זאת ע"י תכנון מערכת מורכבת, הכוללת מודלים מאומנים אשר הציגו פרדיקציות לכל שבוע משחק, ועל פיהם השחקן שמימשנו יבצע את הפעולות הטובות ביותר. המשימה מורכבת מאוד: באיזו אסטרטגיה עלינו לשחק, וכמה "קדימה" עלינו לתכנן את הפעולות שלנו? איך אנו יכולים לתת תוצאות חיזוי ברמת דיוק מספיק גבוה ומהימנה, בהתחשב במחזורים שכבר שוחקו? איך אנו יכולים להביא לידי חשבון מידע "בזמן אמת", אשר לא מופיע לידי ביטוי בנתונים היבשים? לפיכך, החלטנו לפרק את המערכת לרכיבים ותכננו מערכת המורכבת ממספר מודלים. ניתן לחלק את המערכת בצורה גסה לשני רכיבים מרכזיים:

- **מודל חיזוי.** המטרה של רכיבים אלו היא להעמיד לאלגוריתם השחקן את המידע הטוב ביותר שניתן, וגם רכיב זה מתחלק לשני קומפוננטות מרכזיות: מודל חיזוי נקודות ומודל מדיה.

**מודל חיזוי הנקודות** מתבסס על נתונים וסטטיסטיקות מהעונות האחרונות וכן מהעונה הנוכחית – בהתחשב שאנו רוצים לתת הערכה למחזור  $x+1$ , למחזור  $x$  והקודמים לו, שכבר שוחקו ויש בידנו את הנתונים שלהם, יש משמעות אדירה.

**מודל המדיה** מטרתו להעניק מימד מידע נוסף שלא מתקבל מהנתונים והסטטיסטיקות "היבשות". עובדות כמו שחקן שנפצע בסמוך למשחק, מאמן שיוצא בהצרה לתקשורת שבכוונתו להרכיב שחקן מסוים בהרכב, וכן "באז" תקשורתי רחב – יכולות להשפיע רבות על הבחירה. לשם כך, מודל המדיה שלנו, בהתבסס על ציורים ממקורות שונים מטוויטר, מעניק שתי פרדיקציות משמעותיות: כשירות השחקן (האם ישחק) ו-"הייפ תקשורתי". במודל זה השתמשנו בכלים מעולם למידת המכונה ועיבוד שפה טבעית.

- **אלגוריתם השחקן.** הפעולות שהשחקן צריך לבצע בתחילת העונה, ולאחר כל מחזור, מורכבות למדי. הם כוללות וריאציה של בעיית תרמיל הגב עם הגבלות נוספות, תכנון אסטרטגיית חילופים ובחירת השחקנים לכל מחזור וכן אסטרטגיית שימוש בצ'פים – כלים מיוחדים בזמינות נמוכה במהלך המשחק. לכן, מימשנו אלגוריתם אשר מקבל כקלט את תוצאות מודלי החיזוי (צפי נקודות, כשירות והייפ תקשורתי) עבור כל שחקן, ומקבל על פיו החלטה לביצוע פעולות.

המערכת הכוללת מורכבת, והשתמשנו בכלים רבים ומגוונים: איסוף ועיבוד הנתונים, אשר היה מאתגר אך יסודי כדי לאפשר למודלים "קרקע בטוחה" לביצוע ניסויים. בדקנו מספר מודלים שונים, החל מעצי החלטה, דרך רשתות עמוקות כמו RNN ו-CNN. מימשנו כלים ועקרונות מעולם עיבוד השפה הטבעית, כדי לסווג ציורים. כמו כן, השתמשנו כלים אלגוריתמיים לפתרון בעיות NPC, וביצענו Design למערכת שלמה ומרתקת. אך לפני שנצלול לתיאור הבעיה, נתחיל מהיכרות קצרה עם המשחק.

## 2. רקע – פנטזי פריימרליג

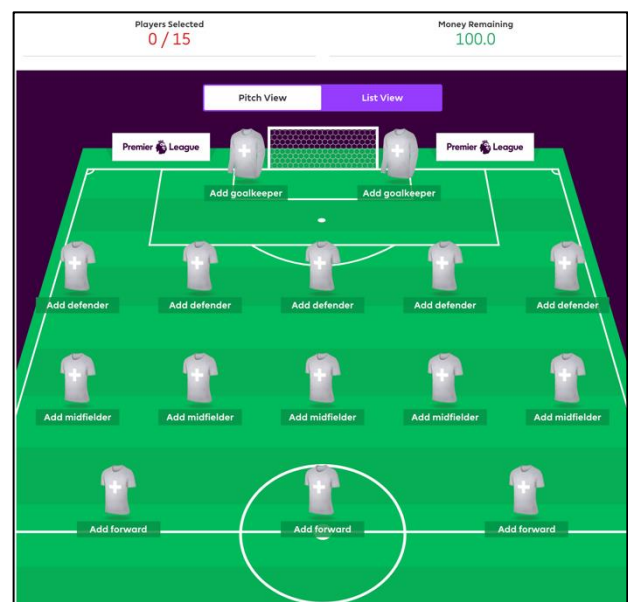
פנטזי פריימרליג הינו משחק אסטרטגיה שבו על המשתמש להרכיב סגל שחקנים דמיוני, והשחקנים שבחר מקבלים ניקוד על בסיס הישגים בפועל בכל מחזור משחק. המשחק משוחק ע"י למעלה מ-8 מיליון משתמשים, בדפדפן ובמובייל, והוא אחד ממשחק הספורט האהובים בעולם (למרות שהוא דורש מינימום פעולות מהמשתמש, אך לא מעט חשיבה ואסטרטגיה!). נציג בקצרה את חוקי המשחק<sup>1</sup> וסימולציה קצרה של משחק ע"י שחקן אנושי, כדי לתת טעימה ומושג קצר על החוקים ומנגנון החלטות שעלינו להתחשב בו בעת ביצוע הפעולות.

### הוראות וחוקים:

- **בחירת סגל ראשוני:** כל משתמש בוחר 15 שחקנים: 2 שוערים, 5 שחקני הגנה, 5 שחקני קישור ו-3 שחקני התקפה. כל שחקן מתומחר בצורה שונה, ולראשות המשתמש 100 מיליון פאונד בתחילת העונה. כמון כן, לא ניתן לבחור יותר מ-3 שחקנים מכל קבוצה.



אהמסך לאחר בחירת הקבוצה. ניתן לראות שניתן לחסוך מעט מהתקציב. כמו כן, למרות שהייתי שמח לבחור יותר משחקני צ'לסי וליברפול (קבוצות חזקות), המגבלות מונעות ממני ולכן בחרתי רק שלושה שחקנים מכל אחת מהן.



מסך בחירת הסגל הראשוני.

- **מחזור משחקים:** כל שבוע מתקיים מחזור משחקים (יש 38 מחזורי משחקים כאלו, המכונים GW). השחקן יכול לבצע פעולות בין כל מחזור משחקים - בחירת הרכב מהסגל הקיים, חילופים אל ומחוץ לסגל ושימוש בצ'יפים מיוחדים. לכל מחזור משחקים יש deadline, שאחריו, לא ניתן לבצע פעולות לאותו המחזור. המשתמש מקבל את הנקודות בהתאם להישגי השחקנים, ע"פ מפתח ניקוד.
- **ניהול ההרכב:** בכל GW, עלינו לבחור 11 שחקנים מההרכב לסגל שלו: שחקנים אלו יקבלו נקודות על סמך הישגים. ארבעת הנותרים נמצאים בספסל. שחקן בספסל צובר נקודות, אך הן אינן נסכמות לניקוד הסופי אלא אם אחד מהשחקנים בהרכב לא משחק בכלל (ואז הספסל הראשון בספסל מחליף אותו, וכך הלאה לגבי כל חיסור). עלינו לעמוד במגבלות מבחינת העמדות: לפחות שוער אחד, 3 מגנים, 3 קשרים וחלוץ. הגבלות אלו מקבעות מספר "מערכים" שבהם ניתן להציב את השחקנים, ולתעדף שחקנים

<sup>1</sup> [החוקים המלאים של המשחק נמצאים כאן](#)

## AI Project - Fantasy Premier League Player

מעמדות מסוימות. כמו כן, על המשתמש להעניק את "סרט הקפטן" לאחד מהשחקנים. שחקן זה יקבל ניקוד כפול באותו המחזור.

- **ניהול סגל וחילופים:** בכל GW, אנו מקבלים אפשרות לבצע חילוף אחד בחינם – כלומר, למכור שחקן מהסגל ולקנות שחקן שמחוץ לסגל בתנאי שאנו עומדים במגבלת התקציב, העמדות וההגבלה על 3 שחקנים מכל קבוצה. אם נרצה לבצע חילוף נוסף, נוכל, אך כל חילוף נוסף שכזה יעלה בקנס של 4 נקודות מהניקוד הכללי של הקבוצה.
- יתרה מזאת, נוכל לא לממש את החילוף החינמי שלנו במחזור מסוים ולצבור אותו למחזור הבא – כך יהיו לראשותנו 2 חילופים ללא קנס. לא ניתן "לשמור חילוף" יותר ממחזור אחד.
- **ניהול תקציב:** כפי שציינו, אנו מקבלים תקציב התחלתי, אך הוא אינו בהכרח נשאר כך. מחירי השחקנים משתנים לאורך העונה, לפי היצע וביקוש. המנגנון של שינוי המחירים מורכב (ואינו מפורסם באופן רשמי ע"י מנהלת הפריימרליג), אך לרוב שחקנים טובים, שצוברים הרבה נקודות, מקבלים עליות מחיר במהלך העונה. כלומר, כל שחקן הוא מאין מניה.
- **צ'יפים:** לראשות המשתמש קיימים צ'יפים, כלומר, יכולות מיוחדות שניתן להפעיל במהלך העונה. לא ניתן להפעיל יותר מצ'יפ אחד בכל מחזור משחק:
  - **Wildcard** – 2 בעונה. הראשון ניתן למימוש מתחילת העונה ועד מחזור 19, והשני ממחזור 19 ועד סוף העונה. ביצוע חילופים בלתי מוגבלים, ללא קנס.
  - **Triple Captain** – הקפטן יקבל ניקוד משולש למחזור אחד.
  - **Free Hit** – החלפת כל סגל הקבוצה למחזור הקרוב. לאחר שהמחזור מסתיים, הסגל המקורי חוזר לידי המשתמש (כלומר, סגל חד פעמי).
  - **Bench boost** – הניקוד של שחקני הספסל שלנו יסכם בניקוד הכללי.
- **שיטת הניקוד:** שיטת הניקוד עבור כל שחקן די מורכבת, ומפורטת עם החוקים המלאים<sup>2</sup>. בכלליות, השחקנים מקבלים ניקוד על בסיס הישגים במגרש, ותלוי העמדה שלהם. כיבוש שער מזכה חלוץ ב-4 נקודות, וקשר ב-5. הישגים בולטים לניקוד: הופעה (2 נק'), שער (4-6 נק'), ביסול (4 נק'), שמירת שער נקי (6 נק'), רק לשוער ושחקן הגנה. לשחקן קישור נק' אחת, עצירת פנדל (6 נק', רק לשוער). כמו כן, בכל משחק 3 שחקנים מקבלים ניקוד בונוס בהתאם להצטיינותם במגרש – הדבר נקבע ע"פ מנגנון מפורט אך מעט מורכב.

### סימולציית משחק ע"י שחקן אנושי לצורך הדגמה – שלושה מחזורים ראשונים:

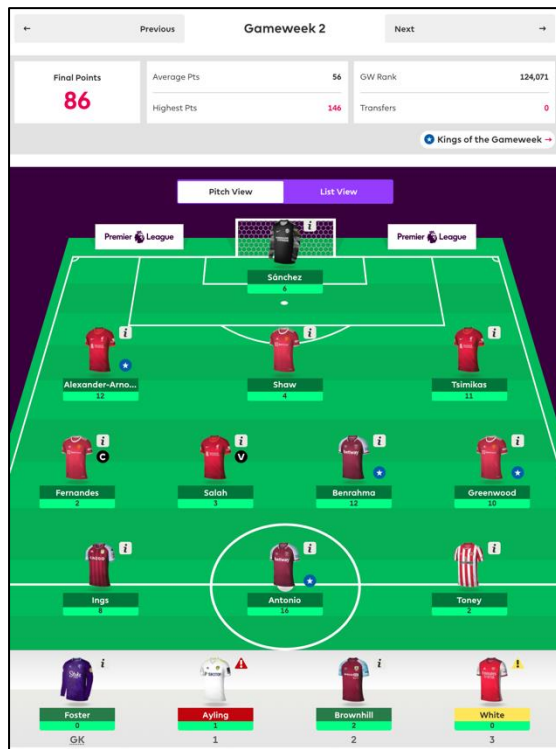
נציג דוגמא קצרה של סימולציית משחק ע"י שחקן אנושי. הדבר כמובן לא מחייב את המודלים והאלגוריתמים דבר, והם מנותקים מכך, אך הסימולציה נותנת אינדיקציה לא רעה על שיטת החשיבה הדרושה של משתמש אנושי. כמו כן, הסימולציה תאפשר הכרות קצרה למשחק ובעיקר לאתגרים המחשבתיים הטמונים בו. במחזור הראשון, בבחירת הסגל, עלינו לבחון את לוח המשחקים של היריבות. כמובן שנעדיף שחקנים איכותיים וחזקים, אך אלו יקרים ואי אפשר לקנות את כולם. לכן, נרצה לתזמן את שחקני ההרכב שלנו עם לוח המשחקים שלהם: נעדיף שחקנים שלהם לוח משחקים קל יותר (כמובן, על הנייר) וננסה להימנע מבחינת שחקנים שלקבוצותיהם רצף משחקים קשה. כמו כן, עלינו לבחון את מחירי השחקנים וביצועיהם בעונות האחרונות,

<sup>2</sup> <https://fantasy.premierleague.com/help/rules>, תחת לשונית ניקוד,

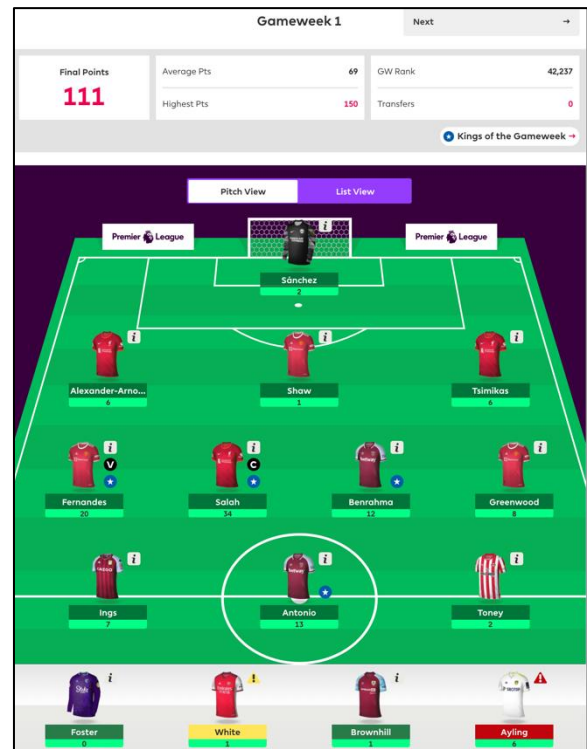
## AI Project - Fantasy Premier League Player

ולנסות להעריך כמה נקודות ישיגו במחזור הקרוב – ואף אחריו! אנו מוגבלים בחילופים, ולכן נשאף לחשוב כמה מהלכים קדימה, כדי לתכנן את התקציב ומגבלות השחקנים שלנו בהתאם לחילופים. אין סוף באמת למנגנון קבלת ההחלטות האנושי. ניתן לנבון שעות רבות בסטטיסטיקות ובנתונים, בלוחות משחקים, ואף לקבל עדכונים לגבי כשירות השחקנים (מי שפצוע או בבידוד בגלל קורונה, כמובן, לא ישחק). המורכבות הזו היא שהופכת את המשחק לכ"כ כיף ומרתק, והרצון להצליח כמה שיותר מדרבן חשיבה וניתוח מעמיק יותר.

במחזור הראשון והשני, נבחר הסגל הבא, אשר הניב 111 ו-88 נקודות בהתאמה בשני המחזורים הראשונים–



הישיג במחזור המשחקים השני. גם כאן, הניקוד גבוה מהמוצע הכללי.

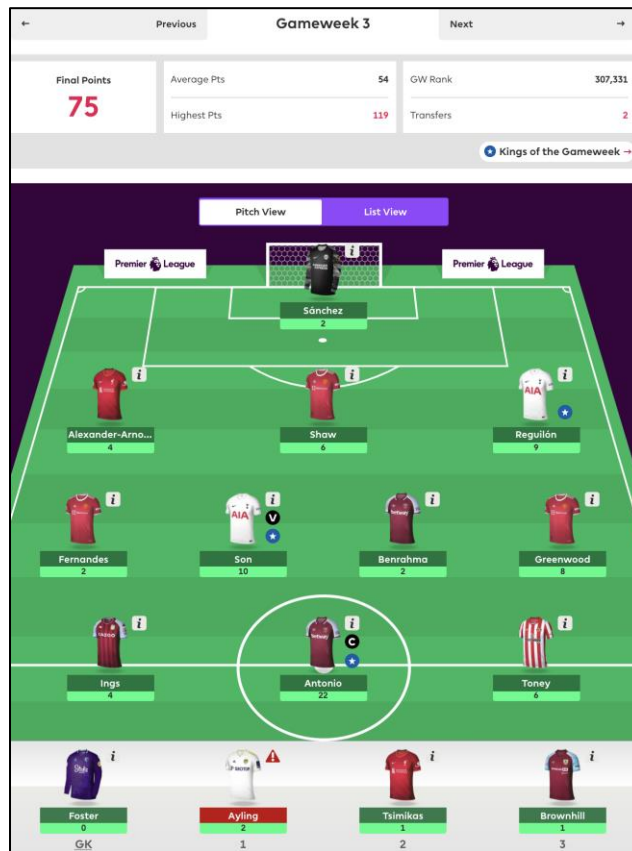


הישיג במחזור המשחקים הראשון. 111 נקודות, ניקוד גבוה מאוד (ניתן לראות את הממוצע של השחקנים הפעילים). כמו כן, כל שחקני ההרכב שותפו ולכן לא בוצעו חילופים.

- **במחזור הראשון:** ראשית, תקציב. הוא אינו נוצל כולו במחזור הראשון. המחשבה שעומדת מאחורי זה שבתחילת העונה יש מספר שחקנים חדשים, שאולי נרצה להכניס אותם בסגל בהמשך ולכן נחסך לכך מעט תקציב. כמו כן, הסגל מראש תוכנן לשימוש מלא וללא חילופים בשני המחזורים הראשונים, כדי לחסוך חילוף למחזור השלישי. ניתן לראות שהקפטן שלנו (סלאח) הניב 34 נקודות, כלומר 17 נקודות בפועל. זה המון למחזור אחד (בהמשך נפרט על ממוצעי ניקוד של כלל שחקני הליגה בכל מחזור. הם די נמוכים מכיוון שהרוב לא משחקים, או משחקים אך בלי הישגים משמעותיים). הדבר נבע מכיוון שבמחזור זה הוא כבש שער אחד, בישל שניים נוספים ונבחר לשחקן המצטיין. המחזור היה די מוצלח, ולכן במחזור השני הוחלט לא לבצע חילוף אלא לחסוך אותו.
- **במחזור השני:** הצלחה נוספת מבחינת הניקוד, למרות שהפעם הקפטן שלנו לא הניב הרבה נקודות. עם אנטוניו היה מסומן כקפטן, היינו יכולים לחגוג עם מחזור נוסף תלת-ספרתי מבחינת הניקוד.

## AI Project - Fantasy Premier League Player

לקראת המחזור השלישי, הוקדשה מחשבה רבה – שחקנים נפצעו או חזרו לכשירות, ולוח המשחקים כמובן נותן את אותותיו. לדוגמא, לטוטנהאם, הקבוצה הלונדונית, הזדמנו משחקים קלים לתקופה הקרובה – לאחר שני משחקי פתיחה קשים במחזור הראשון והשני. לכן, הוחלט לנצל את שני החילופים כדי להכניס שני שחקני טוטנהאם שלהם משחקים קלים יחסית. נבחרו כמובן מי שהנתונים הראו שהם עתידים לשחק בהרכב וכן שיש להם ביצועים טובים. הוחלט להוציא את שחקן ארסנל (וויט), שנפצע, ואת סלאח מליברפול, שהיה נהדר אך במחזור השלישי חיכה לו משחק נגד צ'לסי - שלה הגנה חזקה מאוד. במקומם נכנסו שחקני טוטנהאם, סון ורגיליון, והם אף היו זולים יותר כך שנותר כסף בעתיד אם נרצה להחזיר את סאלח האימתני. ואכן, הבחירות השתלמו! השחקנים שהוכנסו בבשו ובישלו, והמחזור היה מוצלח מאוד:



גם בחירת הקפטן היית משתלמת, ואנטוניו המשיך לכבב וקיבל 11 נקודות (ו-22 כקפטן עם ניקוד כפול).

והאתגר ממשיך. בין מחזור 3 ל-4, התקיימה פגרה. לרוב, בפגרה, שחקנים משחקים בנבחרת שלהם ובתקופת הקורונה הדבר מורכב מאוד: למשל, שחקנים דרום אמריקאים נדרשו לבידוד, ולא היו זמינים במחזור 4. יתרה מזאת, במשחקי נבחרות שחקנים יכולים גם להיפצע ולא להיות זמינים למחזור הליגה שלאחר הפגרה. הסיטואציה הזו לבדה מורכבת, ואולי דורשת שימוש באחד מהצ'פים או אולי לקבל קנס על חילוף כפול כדי לרענן את הסגל.

## AI Project - Fantasy Premier League Player

אך אנו נעצור את הסימולציה הקצרה שלנו במחזור זה. מכאן, הגדרת הבעיה בפרק הבא תראה ידידותיות יותר, ובעיקר הרעיון שעומד מאחורי האתגרים והמשימות לפיצוח בעיה זו.

### 3. הגדרת הבעיה

---

מהם אוסף הפעולות שעל שחקן לבצע בכל מחזור משחק בכדי למקסם את סך הנקודות בעונת מלאה

---

באופן ישיר נגזרת מטרת העל של הפרויקט: מימוש שחקן FPL אשר יצליח לצבור מספר רב של נקודות.

#### הגדרת המעטפת

כדי לבחון את השחקן שנממש, עלינו להריץ אותו על עונת משחקים שלמה. לכן, בחרנו את העונה האחרונה (2020-2021) אשר התקיימה במלואה. לאחר מימוש השחקן במלואו, נבצע סימולציה על עונה שלמה – **כאשר לכל מחזור, עולמו של השחקן תואם את המציאות: קיים לו בכל פרק זמן רק מידע הזמין לאותו תאריך ואותה נקודת זמן, ואין לו שום גישה או מידע למחזורים עתידיים.**

לדוגמא, במחזור המשחקים הרביעי, השחקן שלנו יהיה חשוף לכל המידע שקיים עד לאותו מחזור (לא כולל). לפי מידע זה, וזה בלבד, הוא יקבל את ההחלטות ויבצע את הפעולות הנדרשות. המעטפת כוללת מימוש של "סימולטור משחק". כלומר, לשחקן יש את כל הכלים, האפשרויות וגם המגבלות כמו שחקן אנושי, בהתאם לחוקי המשחק. לפיכך, בכל מחזור השחקן מבצע פעולות כמו בחירת הרכב, ביצוע חילופים, בחירת קפטן והפעלת צ'יפים – ולאחר כל איטרציה של מחזור משחק נקבל אינדיקציה את כמות הנקודות שהשחקן שלנו קיבל בפועל.

#### הגדרת הישגים – האם השחקן שלנו מוצלח?

הגדרת הישגים כאשר יצאנו לדרך הייתה מורכבת – איך נדע האם השחקן שלנו טוב? שיערנו שמכיוון שהמערכת שלנו נבונה, כוללת מודלים מאומנים ומוזנת ע"י דאטה רב, הביצועים צריכים להיות לכל הפחות טובים משל השחקן האנושי הממוצע. לכן הצלחה תימדד אם השחקן שלנו ישיג יותר נקודות בעונת משחקים שלמה משחק ממוצע פעיל (שביצע פעולות לאורך העונה). ע"פ הנתונים<sup>3</sup>, ולאחר סינון שחקנים לא פעילים, ניקוד סופי אשר יוביל אותנו לטופ 2 מיליון בעולם ומעלה יחשב להצלחה מעל השחקן הממוצע. לכן הגדרנו מס' יעדים מבחינת דירוג בסוף העונה: טופ 2 מיליון, טופ 500k, טופ 200k וטופ 100k.

#### מורכבות בבניית השחקן – השערות שעלו טרם הנדסת המערכת

בפרק הרקע נתנו טעימה קטנה על מכלול השיקולים שעלינו לקבל בבניית השחקן:

- השפעת מידע עדכני – האם נספק נאמן מודל חדש לכל מחזור או שנבצע עדכון למודל קיים בזמן אמת? **ההשערה שלנו הייתה שמידע "טרי" יותר יהיה בעל ערך רב יותר.** כלומר אנו יכולים לאמן את המודלים שלנו על נתונים מלפני 3-4 שנים, אך ההשערה שלנו הייתה שככל שהמידע עדכני יותר, כך הוא חשוב יותר.

---

<sup>3</sup> <https://www.anewpla.net/>



## AI Project - Fantasy Premier League Player

לפי השערותנו, לדוגמא עבור מחזור משחק מסוים, הרבה יותר חשוב מה קרה במחזורים הצמודים שלפניו, לעומת מה שקרה בעונה שעברה. בחנו את ההשערה הזו כטווח מרכזי במערכת שמימשנו.

- מידע רב אינו נמצא בנתונים היבשים – שיערנו שהנתונים והסטטיסטיקות הם מעולים, אך אינם מספרים את כל התמונה. אם מאמן של קבוצה מסוימת מודיע במסיבת העיתונאים טרם המשחק ששחקן מסוים לא ישחק, כמובן שאין לנו סיבה לבחור אותו. כמו כן, כשירות שחקן בזמן אמת לא באה לידי ביטוי בנתונים היבשים לקראת כל שבוע. לדוגמא, למחזור 32 אני מקבל את כל הנתונים עד מחזור 32, כולל, אך אם שחקן נפצע באימון המסכם, או חלילה נדבק בקורונה, איך אנו מביאים זאת לידי ביטוי? לנושא זה חשיבות אדירה בעת האחרונה, שכן בשנתיים האחרונות הספורט העולמי בכלל, והכדורגל בפרט, נפגע ומושפע מאוד ממגפת הקורונה.

**ההשערה שלנו הייתה ששילוב מידע שמגיע מהמדיה לגבי שחקנים וקבוצות – פייסבוק, טוויטר, דיווחי חדשות – קריטי להצלחת השחקן שנממש.** המדיה מכילה בתוכה עושר אינפורמטיבי שלא מתקבל מבסיס הנתונים "היבש" כמו עדכוני פציעות וכשירות, או אם יש "הייפ" תקשורתי סביב שחקן מסוים.

- בחירת אסטרטגיה ותכנון עתידי – חלק מהותי במשחק הוא גיבוש אסטרטגיה. כמה מחזורים מראש עלינו לתכנן כדי להכניס שחקן שנרצה? האם עלינו לשמור תקציב לשעת צרה, בהנחה שיפצע לנו שחקן בסגל? איך לבחור את הסגל שלנו – להשקיע רק ב-11 שחקנים חזקים וספסל חלש, או לפזר את התקציב בצורה מאוזנת יותר? ובעיקר, איך ומתי להשתמש בצ'יפים שלנו?

מורכבויות אלו ואחרות מכתיבות את האסטרטגיה ששחקנים שונים נוקטים. זהו לא מצב בינארי של "נכון" או "לא נכון". ניסנו לענות על הדברים המרכזיים במימוש אלגוריתם השחקן, ובעיקר לתת את הכלים ע"י חיזוי של כמה שיותר נתונים רלוונטים, ובחלון זמן קדימה שרואה מעבר למחזור הקרוב.

רשימת אתגרים זו, היא שהובילה אותנו להנדסת המערכת שלנו ולמימוש בפועל של השחקן.

### 4. תיאור המערכת

כדי להגיע להישגים, החלטנו לפרק את המערכת לרכיבים בהתאם למורכבויות שהצגנו בפרק הקודם. לשם כך, מימשנו **סימולטור משחק**, סידרנו **בסיס נתונים** והינדסנו **מערכת השחקן** אשר מורכבת משלושה רכיבים מרכזיים.

- **בסיס הנתונים** הכיל מידע של העונה עליה משחקים, ועוד 2 עונות קודמות. כמו כן, הוקצה החלק מיוחד של ציורים עבור מודל המדיה.
- **סימולטור המשחק** הוא המנוע אשר בוחן את השחקן שלנו. הקלט של הסימולטור הוא המידע האמיתי (כלומר, מה קרה בפועל בכל מחזור – כמה נקודות כל שחקן צבר) וכן את הפלט של מערכת השחקן. הפלט הוא הניקוד של השחקן לכל מחזור בפועל, על בסיס בחירת ההרכב שלו.

ומבחינת מערכת השחקן:

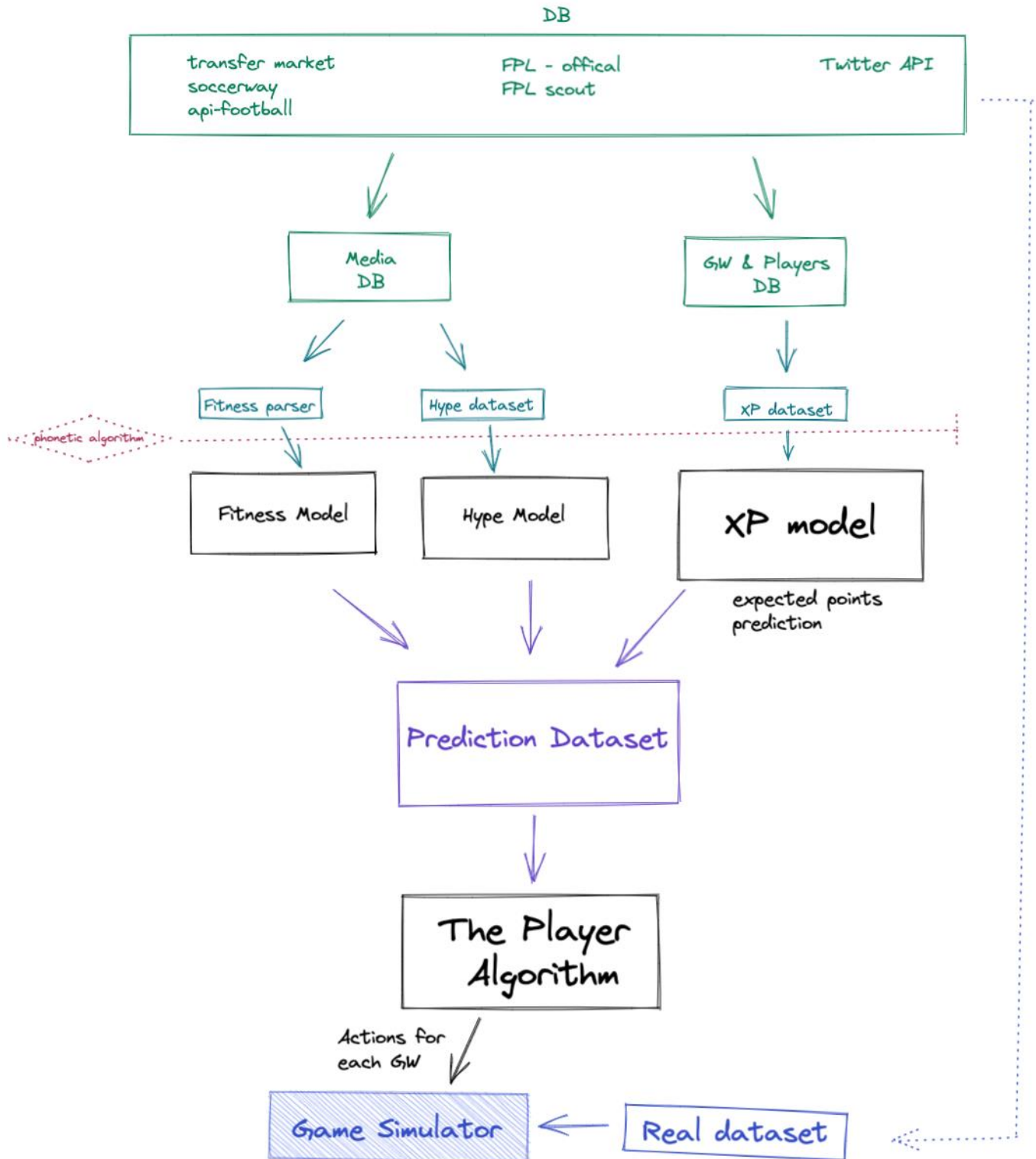
- **מודל חיזוי הנקודות** – המודל המרכזי, אשר נותן חיזוי ניקוד לכל שחקן, בכל מחזור משחק לאורך כל העונה. המודל מבוסס על מידע טבלאי ומסתמך על נתוני העונה הנוכחית ועל העונות הקודמות, והפרדיקציות ניתנות לשלושה מחזורים קדימה. חלק מרכזי במודל זה היה בחינה של שתי גישות: הגישה "הקלאסית" והגישה הדינאמית. האם עלינו לתת יותר חשיבות לדאטה "טרי", כלומר לתת יותר חשיבות למחזורים האחרונים, או לבצע אימון קלאסי על עונות קודמות וממנו את הסיווג? עוד לגבי עניינים אלו בפרק על מודל חיזוי הנקודות.
- **מודל המדיה** – המודל שנותן את הערך המוסף. מוזן מציוצים מטוויטר (premier league sky sports) ונותן חיזוי עבור כשירות השחקן והאם יש לו "הייפ" תקשורתי לקראת מחזור מסוים.
- **אלגוריתם השחקן** – מבצע את הפעולות לפי סט החוקים של המשחק. הקלט שלו הם הפרדקציות, והפלט שלו הם הפעולות (בחירת ההרכב, קפטן, חילוף והפעלת צ'יפים) בכל מחזור משחק.

### אימון, כיוון פרמטרים ובדיקה

**המודל הסופי** מכיל את הרכיבים הנ"ל, מכווננים ע"י הפרמטרים שהביאו לתוצאות הטובות ביותר בשלב הניסויים. חשוב לציין שכל רכיב הונדס בנפרד, נעשו עליו ניסויים ונבחנו מספר מודלים אשר נבדקו אחד מול השני. בפרק "תיאור המערכת" נתאר את המבנים אותם בחנו בכל רכיב, וכן את הרכיב שסיפק את התוצאות הטובות ביותר. בפרק "ניסויים וכיוון הפרמטרים" נפרט על השיטות והבדיקות שביצענו על מנת להגיע לתוצאות הטובות ביותר.

## AI Project - Fantasy Premier League Player

### 4.1 הצגה כללית של המערכת וסימולטור המשחק



## AI Project - Fantasy Premier League Player

### רכיבי המערכת

- **בסיס הנתונים הראשי** הוא נקודת המוצא של המערכת כולה. כפי שיפורט בפרק על בסיס הנתונים, הוא כולל עיבוד ואיסוף של נתונים רבים הקשורים לעולם הכדורגל האנגלי – נתוני כדורגל יבשים ופשוטים, וכן נתונים על משחק הפנטזי עצמו.  
בסיס הנתונים מכאן מתחלק ל-3:
- **GW & Players DB** אשר ישמש את מודל חיזוי הנקודות. כולל מידע טבלאי על העונות האחרונות וגם על העונה עליהם מבצעים את הסימולציה.
- **Media DB** אשר ישמש את מודל המדיה. כולל בתוכו ציורים מחשבוניות טוויטר שונים הקשורים לעולם הפריימרליג (bbc, skysports, וכן חשבונות טוויטר יעודים מעולם הפנטזי).
- **Real Data** אשר מהווה סך הכל נגזרת עבור סימולטור המשחק. מכיל את הנתונים האמיתיים לכל שחקן (שם שחקן ומספר נקודות לכל מחזור).
- **Phonetic algorithm** זהו אלגוריתם השמות. במהלך העבודה על המערכת, זיהנו בעיה מבחינת הצמדת כל שחקן ל-id יעודי. עבדנו עם המון דאטה אשר הגיע ממקורות שונים, ושמות השחקנים נכתבו בצורה שונה – רק שם משפחה לעיתים, כתיב בצורה שונה או שימוש באותיות לטיניות. יתרה מזאת, במודל הטוויטר קיבלנו גם קיצורים וכינויים. לכן, אלגוריתם השמות הפונטיים מבצע "יישור קו" מבחינת שמות השחקנים בכל שמעביר את הטקסט לייצוג הפונטי שלו. כך שם שחקן באותיות אנגליות או לטיניות, אשר בעל אותו צליל, ימופה לאותו מזהה חד-חד ערכי וכך ה-dataset שלנו מיושר עבור כל אחד מהמודלים.
- **expected points – xP Model** - הוא מודל חיזוי הנקודות. לאחר עיבוד הנתונים, משתמש במודלים מעולם הבינה המלאכותית ולמידת הכונה כדי לספק חיזוי נקודות לכל שחקן, עבור כל מחזור.
- **Media Model** הוא כותרת לשני מודלים שונים: מודל הכשירות, אשר מנבא האם שחקן יהיה כשיר או פצוע לכל מחזור משחק, וכן מודל הייפ, אשר נותן ערך לשחקנים בולטים אשר היה סביבם שיח – חיובי או שלילי - ברשתות החברתיות ובעיתונות לקראת כל מחזור משחק.

שלושת המודלים האלו מספקים את הפלט לאלגוריתם המשחק:

מזהה שחקן	קבוצה	xP			כשירות	הייפ
		1	2	3		
מזהה יעודי לכל שחקן.	שם הקבוצה שבה השחקן משחק	צפי למחזור הקרוב, המחזור שאחריו ושניים שאחריו.	צפי למחזור הקרוב, המחזור שאחריו ושניים שאחריו.	צפי למחזור הקרוב, המחזור שאחריו ושניים שאחריו.	ערך בינארי עבור כשיר או לא כשיר.	ערך רציף בין 1 ל-1- עבור האם קיים סביב השחקן הייפ בתקשורת.

כאשר טבלה זו קיימת לכל אחת מהשבועות בעונה שעליה אנו מבצעים את הסימולציה.

עד כאן שלב בניית הפרדיקציות. בסופו, קיים לאלגוריתם השחקן הטבלה לעיל **לכל שבוע משחק**. כמובן שהשחקן אינו חשוף לפרדיקציות עתידיות ובטח שאינו חשוף לדאטה האמיתי של אותו השבוע.

- **אלגוריתם השחקן** מקבל את הקלט הנ"ל, ומבצע חישובים ואסטרטגיות לקראת בחירת הפעולה. הפעולות שהשחקן יכול לבצע בכל מחזור הם בהתאם לחוקי המשחק והתקציב: בחירת הרכב, ביצוע חילופים חיצוניים, בחירת קפטן והפעלת צ'יפ. הפלט הוא ההרכב הנבחר, הקפטן והספסל.
- **סימולטור המשחק** מסמלץ עונת משחקים שלמה: בכל שבוע משחק, מקבל את הפלט של אלגוריתם השחקן ומחזיר את הניקוד אשר הישג באותו שבוע. בסופו של דבר, הסימולטור פולט את ניקוד השחקן המצטבר בסוף העונה.

## AI Project - Fantasy Premier League Player

### 4.2 בסיס הנתונים - בנייתו ועיבוד מקדים

לאחר שלב התכנון, ניגשנו למלאכת איסוף ועיבוד הנתונים. בסיס הנתונים שלנו התחלק לשני חלקים מרכזיים: הדאטה הטבלאי, אשר כולל נתונים יבשים/גולמיים: סטטיסטיקות ונתונים שקשורים לכדורגל, כמו שערים, בישולים, שמירת שער נקי, דקות משחק ועוד. בנוסף, כללנו גם נתונים הקשורים למשחק – למשל, כמה משתמשים הכניסו והוציאו שחקן מהסגל שלהם באותו מחזור, כמה נקודות בונס השיג שחקן מסויים ועוד. החלק השני של בסיס הנתונים כלל את הציורים מטוויטר. בחרנו להשתמש בטוויטר מכיוון שהוא מכיל איגוד של פלפורמות שונות: עיתונות קלאסית, כמו הערוצים הפופולריים bbc ו-sky, שלהם היו ערוצי פריימרליג יעודיים ופעילים. כמו כן, טוויטר מכיל מאות חשבונות טוויטר אשר עוקבים ומדברים על עולם הפנטזי, וחשבנו שנוכל להעזר בחלק מהחשבונות הללו.

נציין ששלב זה כלל ריכוז ובניה של הדאטה-בייס, אך לא כלל עיבוד משמעותי לפני: שלב זה יפורט בחלק המודלים.

#### הדאטה הגולמי – נתונים וסטטיסטיקות כדורגל ופנטזי

החלטנו לאסוף דאטה עבור 3 עונות: עונת אימון (2018-2019), עונה עבור טסטים וכיוון פרמטרים (2019-2020) ועונה עבור סימולטור המשחק (2020-2021).

מסד הנתונים מכיל בקרוב 62,700 רשומות, בצורה הבאה:

$$3_{seasons} * 38_{gw} * 550_{players} = 62,700$$

כאשר מס' השחקנים הוא בממוצע ומשתנה בין העונות (שחקנים פעילים בכל זמן נתון)

הדאטה הגולמי חולץ מהמקורות הבאים: fpl official, soccerway, anewpla. לאחר חילוץ, סודר במבני ההיררכי הבא שבאיוור משמאל. הוא כולל נתוני כדורגל (דק' משחק, שערים וכו') וגם נתונים הקשורים למשחק עצמו (כמה בחרו את השחקן לקפטן? כמה הכניסו והוציאו אותו מהסגל שלהם? וכו')

כל עונה מכילה את הנתונים עבור כלל המחזורים. כל מחזור, מכיל 500-600 רשומות, עבור כל שחקן והנתונים הסטטיסטיים המתאימים. לדוגמא, העמודות הראשונות עבור 13 שחקנים מהמחזור השני של עונת 20/21:

	code	name	assists	bonus	bps	clean_sheets	creativity	element	fixture	goals_conceded	goals_scored	ict_index	influence
0	233425	Aaron Connolly	0	0	2	27	1	11.3	78	16	0	1	6.9
1	55459	Aaron Cresswell	0	0	0	17	0	35	435	9	2	0	6.2
2	74471	Aaron Mooy	0	0	0	0	0	0	60	16	0	0	0
3	225321	Aaron Ramsdale	0	0	0	10	0	0	483	10	1	0	1.1
4	214590	Aaron Wanless	0	0	0	0	0	0	313	15	0	0	0
5	121599	Abdoulaye Doucoure	0	0	0	13	0	2.7	512	12	2	0	0.9
6	197030	Aboubakar Keita	0	0	0	3	0	2	190	13	4	0	0.5
7	159533	Adama Traoré	0	0	0	17	0	39.2	465	18	3	0	6.7
8	80179	Adam Forsyth	0	0	0	0	0	0	199	13	0	0	0
9	39155	Adam Lallan	0	0	0	3	0	1.2	54	16	0	0	0.8
10	110735	Adam Webster	0	0	0	21	1	1	66	16	0	0	1.1
11	46483	Adrien Silva	0	0	0	0	0	0	220	14	0	0	0
12	449926	Adrián Panadero	0	0	0	0	0	0	537	18	0	0	0
13	60706	Adrián Panadero	0	0	0	0	0	0	245	11	0	0	0

לאחר המיזוג לטבלה אחת, איחדנו גם את הפלט של אלגוריתם השמות (יפורט בפרק בהמשך) וכן את קוד השחקן – מזהה חד חד ערכי. סה"כ לכל שחקן קיימים 38 עמודות (פיצ'רים לכל שבוע). 3 מהם מתייחסות לזיהוי שלו (שם, שם פונטי ו-id) כך שמדובר ב-35 נתונים עבור כל שחקן. נציג בצורה מדגמית משמעות של חלקם:

Goal_conceded	Yellow cards	Clean sheets	Goals	Assists	Name_p	Code
שערים שספגה קבוצתו	כרטיסים צהובים שספג (0-2)	שערים נקיים (רלוונטי למגנים ושוערים)	שערים שבבש	בישולים שביישל	הפלט של האלגו הפונטי	מזהה השחקן
Was home	Price	Position	opponent_team	Team	Transfer balance	ict_index
האם לשחקן זה היה משחק ביתי.	מחיר במשחק	עמדה (שוער, מגן, קשר או חלוץ)	היריבה באותו מחזור משחק	קבוצה	מאזן חילופים בקרב כלל שחקני המשחק	מדד של lqi עבור יצירתיות והשפעה. ערך רציף חיובי.

המידע הנ"ל מזין את מודל חיזוי הנקודות בשלבי האימון והולידציה (עבור כיוון הפרמטרים).

## AI Project - Fantasy Premier League Player

### דאטה מטוויטר - מדיה

רצינו לממש שני מודלים, ולשם כך נדרשנו לציוצים רבים לאימון ולאחר מכן לשלב הפרדקציה. **עבור מודל הכשירות**, מצאנו חשבון טוויטר מעולה למשימה<sup>4</sup> אשר מציג בצורה שיטתית, בעלת מבנה קבוע, את כשירות השחקנים באופן רציף. לכן, הוחלט שאין צורך במודל למידה כאשר המבנה של הטקסט "רובוטי" ולכן פשוט מימשנו parser אשר מנקה את המידע הדרוש.

דוגמא של 10 ציוצים טרם העיבוד, בדגש על הציוץ עצמו:

created_at	full_text
Mon May 24	#FPL Update: Edouard Mendy - Rib Injury #CFC Expected Return: 29-05-2021 Status: 50% <a href="https://t.co/xBkirEucuv">https://t.co/xBkirEucuv</a>
Mon May 24	#FPL Update: N'Golo Kante - Tight Hamstring #CFC Expected Return: 29-05-2021 Status: 75% <a href="https://t.co/xBkirEucuv">https://t.co/xBkirEucuv</a>
Mon May 24	#FPL Update: Florin Andone - Tight Quadriceps #BHAFC Expected Return: 01-06-2021 Status: Ruled Out <a href="https://t.co/xBkirEucuv">https://t.co/xBkirEucuv</a>
Mon May 24	#FPL Update: Nathan Ferguson - Achilles Tendon Rupture #CPFC No Return Date Status: Ruled Out <a href="https://t.co/xBkirEucuv">https://t.co/xBkirEucuv</a>
Sun May 23	#FPL Update: Joao Cancelo - Sending Off - Red Card #MCFC Expected Return: 24-05-2021 Status: 100% <a href="https://t.co/xBkirEucuv">https://t.co/xBkirEucuv</a>
Sun May 23	#FPL Update: Joshua King - Lower Back Pain #EFC Expected Return: 01-06-2021 Status: Ruled Out <a href="https://t.co/xBkirEucuv">https://t.co/xBkirEucuv</a>
Sun May 23	#FPL Update: Moussa Sissoko - Knock #COYS Expected Return: 01-06-2021 Status: Ruled Out <a href="https://t.co/xBkirEucuv">https://t.co/xBkirEucuv</a>
Sun May 23	#FPL Update: Tanguy Ndombele - Knock #COYS Expected Return: 01-06-2021 Status: Ruled Out <a href="https://t.co/xBkirEucuv">https://t.co/xBkirEucuv</a>
Sun May 23	#FPL Update: Giovanni Lo Celso - Knock #COYS Expected Return: 01-06-2021 Status: Ruled Out <a href="https://t.co/xBkirEucuv">https://t.co/xBkirEucuv</a>

ניתן לראות שכל ציוץ בנוי ממבנה מסוים בחשבון זה, ולכן היה פשוט יחסית לפרש את הציוצים ולעבד אותם פירוט על מימושו בפרק מודל המדיה.

**עבור מודל ההייפ**, החלטנו להתמקד בחשבון של [@skySportPL](https://twitter.com/skySportPL)<sup>5</sup> אשר מכיל, נכון לכתיבת שורות אלו, מעל 190 אלף ציוצים (!) אשר כולם קשורים לפריימרליג האנגלית. מכיוון שקיימת הגבלה מובנת ב-API של טוויטר עבור שליפת ציוצים, נאלצנו לשלוק ציוצים גם מחשבונות אחרים עבור שלב האימון. דאגנו שהחשבונות יהיו בעלי אופי דומה – כלומר, עיתונאי ומהימן ([@BBCSports](https://twitter.com/BBCSports), [@Andy](https://twitter.com/Andy), [@premierleague](https://twitter.com/premierleague)).

דוגמא של 10 ציוצים טרם העיבוד:

created_at	id	id_str	full_text	contributors	is_quote_status	retweet_count	favorite_count	favorited	retweeted	lang	entities.hashtags	user.name	user.screen	user.location
Fri Sep 17 12:16:00 +0000 2 1E+18	1.4388E+18		"He's absolutely the best coach in the world - but		FALSE	49	814	FALSE	FALSE	en	[]	Sky Sports P	SkySportsPL	London
Fri Sep 17 11:40:00 +0000 2 1E+18	1.4388E+18		"Pep Guardiola has started the Spaniard in all six		FALSE	24	537	FALSE	FALSE	en	[]	Sky Sports P	SkySportsPL	London
Fri Sep 17 11:27:05 +0000 2 1E+18	1.4388E+18		"Our supporter base is different -		FALSE	46	426	FALSE	FALSE	en	[{"text": "MCFC", "indices"	Sky Sports P	SkySportsPL	London
Fri Sep 17 11:04:00 +0000 2 1E+18	1.4388E+18		"We've got some women that work with the tea		FALSE	26	1048	FALSE	FALSE	en	[]	Sky Sports P	SkySportsPL	London
Fri Sep 17 10:22:00 +0000 2 1E+18	1.4388E+18		#WHUFC, #LCFC and #THFC were		FALSE	14	395	FALSE	FALSE	en	[{"text": "WHUFC", "indice	Sky Sports P	SkySportsPL	London
Fri Sep 17 09:39:00 +0000 2 1E+18	1.4388E+18		How much is Jude Bellingham worth? üü		FALSE	156	4980	FALSE	FALSE	en	[]	Sky Sports P	SkySportsPL	London
Fri Sep 17 09:01:00 +0000 2 1E+18	1.4388E+18		"If I thought another way of playing would impro		FALSE	14	499	FALSE	FALSE	en	[]	Sky Sports P	SkySportsPL	London
Fri Sep 17 08:29:08 +0000 2 1E+18	1.4388E+18		How do you solve a problem like		FALSE	34	614	FALSE	FALSE	en	[{"text": "GameZero", "inc	Sky Sports P	SkySportsPL	London
Fri Sep 17 08:02:00 +0000 2 1E+18	1.4388E+18		Can Tottenham bounce back from their 3-0 defea		FALSE	27	430	FALSE	FALSE	en	[]	Sky Sports P	SkySportsPL	London

ניתן לראות שפרט לציוץ עצמו, אנו מקבלים מידע נוסף – כמה אהבו אותו, האם הציוץ הוא retweets, את רשימת התיוגים שלו ועוד.

- מעל 15,000 עבור האימון שמתוכם קצת מעל 4,500 עברו את הסינון והעיבוד והיוו את האימון.
- מעל 9,600 עבור שלב הסיווג. מתוכם רק 487 עובדו, סוננו ועברו את הסיווג עצמו.

עיבוד הציוצים, סיווגם לאימון והכנתם לקראת המודל מפורטים באופן מלא בפרק על מודל המדיה.

<sup>4</sup> <https://twitter.com/PremierInjuries>

<sup>5</sup> <https://twitter.com/SkySportsPL>

### 4.3 Expected Points Model – חיזוי נקודות ע"פ סטטיסטיקות ונתוני המשחק

כפי שהוסבר לעיל, בעיה מרכזית אותה נדרשנו לפתור היא בעיית Regression שתפקידה לחזות עבור כל שחקן במשחק כמה נקודות הוא עתיד להניב במחזורים הקרובים. כשניגשנו לפתרון בעיה זו התלבטנו בין שתי גישות מרכזיות לפיהן ננסה לבצע את הפרדיקציות – האם להתייחס למידע באופן קלאסי, או כ"מידע נושם". Spoiler alert: הגישה השנייה הניבה ביצועים טובים יותר, והתוצאות הסופיות מבוססות עליה כמפורט בהמשך.

לכל גישה יתרונות וחסרונות, כמו גם קשיים טכניים ולוגיים שהועלו בזמן התכנון ותוך כדי ריצה. לשתי הגישות קיימים גם קשיים חופפים, הנובעות מהמטרה שהוגדרה שהיא בבסיסה חיזוי פעולות אנוש. מתוקף היותנו בני אדם, מאורעות רבים משפיעים עלינו באופן זה או אחר – ולא מעט מאירועים אלו לא ניתן לכמת בצורה טובה מספיק, או שלא ניתן לכמת בכלל. ביצועי השחקנים מושפעים גם מכך ששחקן מסוים יכול "לתפוס יום" (לחיוב או לשלילה), החלטות שיפוט שונות, מתח בחדר ההלבשה, שיפוץ בבית השכן וכן הלאה. כמובן שמדובר בשחקנים מקצוענים ולכן השפעתם של אירועים אלו אמורה להצטמצם, אך אפילו יניב קטן הגדול לא היה במיטבו כאשר לא ישן טוב, רב עם אישתו, נפצע או סתם היה ביום חסר מזל. על מנת להתגבר על חלק ממכשולים אלו, ניסינו כאמור לבחור מדדים המשקפים את הציפיות האלו – כמו גם הוספת מודלים שנועדו להשלים את התמונה (פציעות, Hype – יפורט בהמשך) ולספק לאלגוריתם הסופי תמונה מלאה יותר בעת בחירת ההרכבים וביצוע החילופים.

היות והמידע שהוצאנו עבור כל שבוע כלל באופן טבעי לא מעט מביצועי השחקנים באותו מחזור, נתונים שמן הסתם אין ברשותנו טרם התרחש המחזור, החיזויים בוצעו על סמך ביצועי השחקנים במחזורים הקודמים. דבר זה הוביל לקושי משמעותי במחזור הראשון בכל עונה, שכן הוא... ראשון. על מנת להתגבר על מכשול זה, הוספנו ידנית מחזור טרום-עונה אשר מתבסס על ממוצעי שלושת המחזורים האחרונים של השחקנים הרלוונטיים בעונה הקודמת, תוך מחשבה שהמשכיות מסוימת תשמר.

נפרט כעת על שתי הגישות, ולאחר מכן נציג את המודל שבבחר.

#### 4.3.1 גישה סטטית

בגישה זו התייחסנו ל-Data באופן "קלאסי", משמע ביצוע פרדיקציה על סמך התכונות שחולצו ללא התייחסות למועד בו התרחשו המחזורים. בגישה זו יצאנו מנקודת הנחה שהקשרים בין הנתונים היבשים לתוצאות אינם תלויים במימד הזמן, וסט תכונות מסוים ייתן אינדיקציה טובה ללא תלות בזמן בו הוא נדגם. ניתן להסתכל על גישה זו בתור ניסיון לחזות את כל העונה מראש – על סמך הקשרים שהוסקו מנתוני שתי עונות האימון, ננסה להסיק מה יקרה בעונת המבחן.

גישה זו אינה מבחינה בין עונות ומחזורים שונים בתהליך הלמידה וההסקה, אלא בוחנת סט תכונות מסוים בפני עצמו, וללא תלות ב"הקשר" ממנו הוא נלקח. תהליך זה טוב ויפה כאשר הרשומות ב-Dataset בלתי תלויות זו בזו, או לפחות כאשר התכונות הנבחרות משקפות את התוצאה של תלויות אלו בצורה שלמה מספיק. תוך כדי הניסויים, עלה חשד כי התכונות שנבחרו אינן מספקות בצורה שלמה דיה את תוצאות הקשרים לצרכים אלו, ולכן החלטנו להוסיף תכונות בדמות סטטיסטיקות עבר (מפורט בפרקים הקודמים). עלו גם קשיים טכניים נוספים, כמו חוסר אחידות רציני בהתפלגות התוצאות של השחקנים השונים – כפי שניתן

## AI Project - Fantasy Premier League Player

לראות בהיסטוגרמה בפרק על [בסיס הנתונים](#). על קושי זה ניסינו להתגבר במספר דרכים, כמו איזון כפוי של ה-Dataset ע"י השמטת מאורעות בעלי ניקוד 0, וכן מתן משקל יתר לעצים בעלי יותר מאורעות עם ניקוד שאינו 0. חשוב לציין שקושי זה עלה גם בגישה השנייה, אך הוא היה פחות משמעותי היות ובכל איטרציה כמות הרשומות שם הייתה קטנה משמעותית.

גישה זו הרגישה לחלקנו יותר נכונה בהיבטי Data נטו, אך סבלה מקושי מהותי באופיה – היא אינה מבחינה בצורה ברורה מספיק בין שחקנים "חמים" ל"קררים", ואינה מתייחסת דיה לשינויים שעוברים על השחקנים, כמו גם על הליגה, השחקנים הנוספים ב-Fantasy, האנליסטים שהפיקו את הסטטיסטיקות וכו'.

### 4.3.2 גישה דינאמית

בגישה זו התייחסנו ל-Data באופן "דינאמי", בו התייחסנו לנתונים כ-Stream חי של מידע והחיצוי התבסס על "העבר הקצר". בגישה זו יצאנו מנקודת הנחה יותר אנושית לתחום הספורט, והיא שלמומנטום יש משמעות ושחקנים מושפעים יותר ממשחקיהם האחרונים מאשר מאלו שהתרחשו שנתיים קודם לכן. ניתן להסתכל על גישה זו בתור "למידה תוך כדי תנועה", בה המודל מתעדכן בזמן אמת ומתאים את עצמו למחזורים האחרונים שהתרחשו.

גישה זו מורכבת יותר, שכן היא כוללת מורכבות נוספת של עדכון תוצאות בזמן אמת: כאשר אנו באים לחזות 3 מחזורים קדימה, כיצד נתייחס לסט הנתונים שהגיע זה עתה? גישה זו דורשת עדינות רבה יותר בטיפול במידע, שכן אילו היו לנו את תוצאות האמת של עוד שבועיים, כל הפרויקט היה מיותר (כמו גם ה-Fantasy עצמו, והווינר...). מורכבות נוספת בבחינת ארכיטקטורת המודל היא כמות מחזורי העבר עליהם יש להתבסס – מצד אחד, הסתמכות על מספר מצומצם של מחזורים ייתן תחזית יותר "חמה". מצד שני, ככל שמסתכלים על יותר מחזורים כך המודל מושפע פחות מאירועים חריגים (ולא חסרים כאלה, בלי עין הרע) ומסתמך על Dataset רחב יותר. ראוי לציין כי ביצענו ניסויים תוך שימוש במודל ממשפחת ה-RNN (Recurrent Neural Network), אשר על אופיו מפורט בהרחבה בפרק [Media Model](#), בו הקושי האחרון מתבטל באופן טבעי. אך כפי שיוסבר בהמשך, נראה כי מודל זה אינו מתאים לתרחיש שלנו.

### 4.3.3 תיאור המודל הנבחר

בנינו מודל אשר נותן פרדיקציה עבור 3 מחזורים קדימה, כאשר עבור כל טווח חוזי (מחזור נוכחי, הבא, וזה שאחריו) קיים תת מודל אשר אומן למטרה זו בלבד – כלומר תת מודל אחד נותן פרדיקציה למחזור אחד בלבד. כל תת מודל שכזה, אשר בגרף מטה נקרא RandomKNNForest, מורכב מ- $N$  עצי החלטה אשר קיבלו Slice שונים מה-Datasetים המתאימים לטווח החוזי (מסומנים בעיגול באיור מטה). החלוקה לעצים אלו מתבצעת ברמת השחקנים כאשר כל תכונה מנורמלת בנפרד, וכן ערכי ה- $N$  הותאמו לכל תת מודל. בעת ביצוע פעולת חיזוי (Predict) מתבצעת פניה לכל אחד מה-RandomKNNForests, אשר מחפש את  $K$  עצי ההחלטה הקרובים ביותר לסט התכונות (שעבר נרמול לפי אותן משקולות) – כאשר המרחק מחושב לפי נורמת  $L_2$  ומתייחס לממוצע המנורמל של ערכי העץ (Centroid) כנקודת הייחוס. הקלט הוא תוצאות השחקנים במחזור הקודם, והפלט הוא התחזית לציון שיתקבל עבור שלושת המחזורים הקרובים, כאשר כל תוצאה מתקבלת מתת מודל אחר.

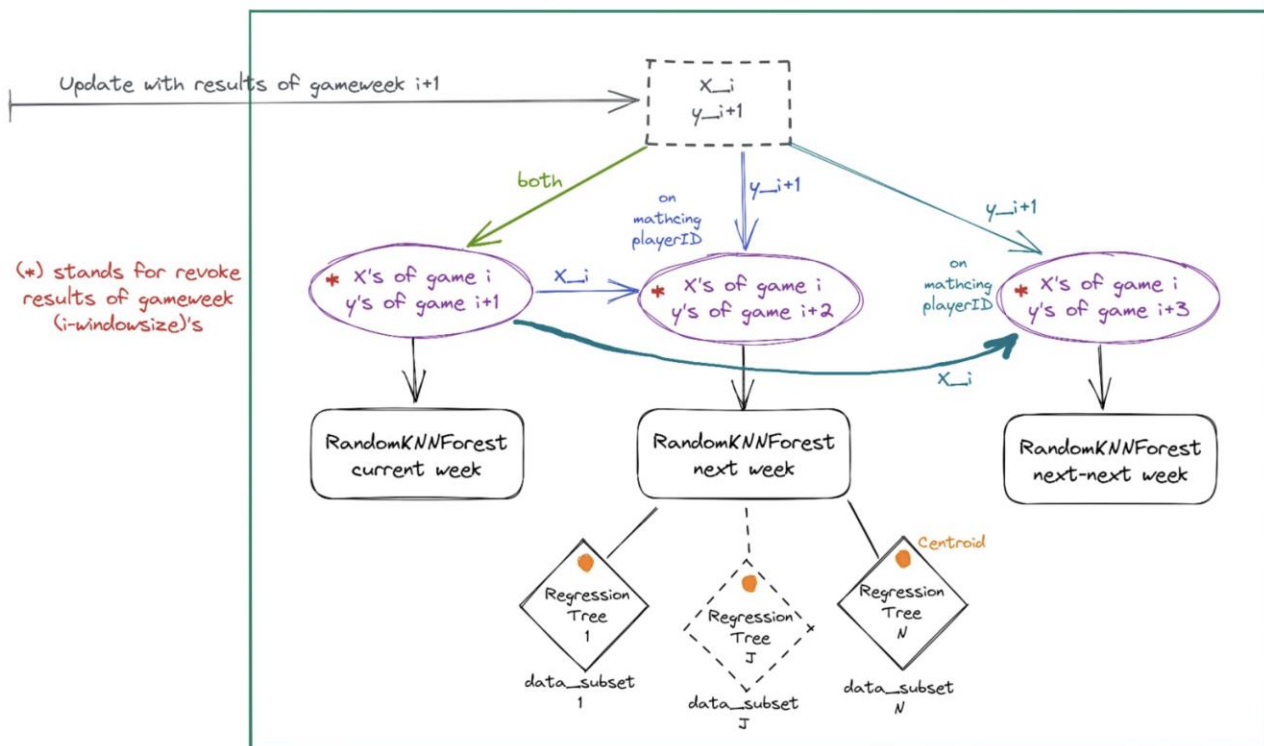
עד כה זה טוב ויפה, ודומה למה שנקטנו בגישה הקלאסית – אך רצינו להתייחס לסיטואציה בה המודל מתעדכן בזמן אמת. למטרה זו הוספנו מנגנון היסטוגרמות חכם אשר שומר בעבור  $WinowSize$  המחזורים האחרונים את



## AI Project - Fantasy Premier League Player

ביצועי השחקנים והתוצאות במחזורים העוקבים, ויודע לייצר לכל אחד מה-RandomKNNForesets את ה-Dataset המתאים לטווח החוזי המתאים לו. בתום כל מחזור, עלינו לעדכן את המערכת ע"י פסילת מחזורים החורגים מגודל החלון שהוקצה וכן להוסיף את התוצאות של המחזור העדכני. באיור המצורף מטה, המתאר בצורה סכמתית את המודל, ניתן לראות גם את תהליך העדכון: בעת קבלת  $X_i, y_{i+1}$  התוצאות של מחזור  $i + 1$  והתכונות של מחזור  $i$  הקדם לו, אנו מעדכנים את ה-Dataset של תת המודל החוזה מחזור אחד קדימה, ובהתאמה למידע השמור בו מעדכנים גם את ה-Dataset של תתי המודלים האחרים (מסומנים בסגול) – שכן כעת יש לנו מבט רחוק יותר אל העתיד עבור רשומות אלו (ניתן להתאים בין הרשומות של המחזור הקודם לבין תוצאות המחזור הנוכחי), תוך ערבוב סדרי הרשומות במטרה לייצר עצים המורכבים מכמה שיותר מחזורים. נקודה זו דיי עדינה עקב כך שזהות השחקנים המשתתפים כל שבוע אינה קבועה, ולכן מתבצעת השוואה על סמך מזהה השחקן כדי להמנע מחוסר תיאום בתוך ה-Dataset. לבסוף, אנו מסירים משלושת ה-Datasets את אירועי המחזור הותיק ביותר שאינו עומד ב-WindowSize שהוגדר.

### Sliding Windows Regressor Forest



עבור מודל זה השתמשנו בשלושה קונספטים עיקריים מעולמות הבינה מלאכותית:

1. Regression Tree: היחידה הבסיסית ביותר במודל, ובין הבסיסיות בתחום. עץ החלטה פשוט הנועד לפתור בעיית רגרסיה על חלקי המידע שניתנו לו.
2. KNN: המודל המוכר והאהוב משיעורי בינה מלאכותית, אשר מנסה לבבא את התוצאה המתאימה ביותר עבור איבר בעל וקטור תכונות  $X$  תוך כדי מיצוע (ממשוקל או שלא ממשוקל) התוצאות שנחזו בעבר עבור  $K$  האיברים הקרובים ביותר ל- $X$ . כוחו בא לכדי ביטוי במודל זה בכך שלרוב לשחקנים בעלי מדדים קרובים יהיו

## AI Project - Fantasy Premier League Player

ביצועים קרובים יחסית, וזה בדיוק מה ש-KNN עושה. השתמשנו כאן בקונספט זה עם עצי רגרסיה במקום מאורעות על מנת להעצים את יכולת ההכללה של המודל.

3. Random Forest: זהו הקונספט שלפיו ביצענו את החלוקה הכללית של תת המודל – כל עץ קיבל סט תכונות אשר מתייחס רק לחלק מהשחקנים, ולפיו חושב הפתרון. חלוקת השחקנים בצורה רנדומית תרמה לכך שכל עץ יהא מגוון יותר, משמע ההחלטה תתקבל על סמך הסתכלות על מאורעות הטרוגניים ככל הניתן. בשילוב עם חישוב המרחק לפי Centroid של תתי העצים, קיבלנו מודל אשר מצד אחד בעל יכולת הכללה טובה ו-Overfitting נמוך, ומצד שני עדיין נותן חשיבות גבוהה יותר למאורעות דומים.

כפי שצוין לעיל, בוצע ניסיון לחיזוי באמצעות RNN. זוהי ארכיטקטורת למידה עמוקה אשר בה קיים "משווא חוזר" של המידע שנכנס, כך שהנתונים של מחזור  $i$  ישפיעו על חישוב הפלטים עבור מחזור  $i + 1$ . על הנייר, נשמע אידיאלי – המודל ילמד את המשקולות הרלוונטיות עבור כל שחקן, את הקשרים הספציפיים שבין הסטטיסטיקות שלו לבין ביצועיו בפועל, ובעצם יהווה מערכת למידה עמוקה אשר נועדה By design להתמודד עם מידע אשר מתעדכן באופן שוטף. ובכן, מסתבר כי ה"ל נכון (אם כי מורכב מאוד לאימון, כפי שמצוין בספרות לא פעם) – אך ארכיטקטורה זו נועדה בעיקרה לחיזוי Time-series data כמו אותות מסנסור, טקסט או וידאו – בדגש על מקור מידע יחיד. במודל זה המצב שונה – אנו מקבלים בכל מחזור מידע לא ממקור יחיד, אלא למעשה מלמעלה מ-500 מקורות בו"ז: כל שחקן (למעשה רוב השחקנים, אך אין זה משנה יותר מדי את המסקנה) בהקשר זה הוא בעצם מקור מידע נפרד, שכן בעיקרון מוחמד סלאח מליברפול מתפקד במשחקו מול ווסטהאם באופן בלתי תלוי בביצועיו של אלכסיס סאנצ'אז מארסנל (כיום, משחק בפנרבחצ'ה בליגה הטורקית) במשחקו מול לידס. אנו מאמינים כי ניתן להתגבר על מכשול זה תוך העמקת הידע ב-RNN, ולהוביל לכך שהמודל יניב ביצועים העולים על אלו שהשגנו בפוייקט זה – שכן הפוטנציאל בלמידה סדרתית, עם רכיב זיכרון, קיים.

בסופו של דבר, הפלט של רשומה אחת ממודל חיזוי הנקודות הינו:

Player id	Ps1	Ps3	Ps3	GW
מזהה שחקן ייחודי	הניקוד החזוי עבור המחזור הבא, המחזור שאחריו ושני מחזורים אחריו. הטווח הוא רציף, ללא הגבלה מסויימת. אצלנו, טווח החיזוי היה בין 0 לבין 11 נקודות.			המחזור שהפרדקציה רלוונטית עבורו

### Media Model 4.4 – חיזוי כשירות ו"הייפ"

מטרת מודל המדיה היא לתת מידע נוסף, שלא בא לידי ביטוי בנתונים הסטטיסטיים ונתוני המשחק. סטטיסטיקות מעולות לחיזוי הנקודות, אך אינן יודעות להתמודד עם מאורעות שקרו "בין משחקים", ודברים שלא מתקיימים על המגרש או קשורים באופן ישיר למשחק הפנטזי.

הדבר מהותי בנושא **כשירות וזמינות שחקנים**. אם שחקן נפצע בין מחזורי משחק, חלה בקורונה או אפילו נכנס לבידוד – לא נקבל את זאת מהנתונים היבשים. עלינו לפנות למקורות מידע אחרים: התקשורת והרשתות החברתיות, כדי להבין מה מצב כשירות השחקנים.

יתרה מזאת, חלק קריטי במשחק הפנטזי הוא לתפוס את השחקן הנכון, בזמן הנכון. התזמון הוא כלי חשוב: אם נצליח להביא שחקן שעומד "להתפוצץ" ולספק ניקוד רב, למרות שהדבר לא מופיע בנתונים היבשים, נצליח לצבור פער על משתמשים אחרים בניקוד שלא השכילו להכניס אותו לסגל שלהם. **ההייפ סביב שחקן הוא מהותי**, והדבר לא בא לידי ביטוי בנתונים היבשים. חיזוי הנקודות מסתמך אך ורק על **משחקים ששוחקו והישגי השחקנים**. אך האם זה הדבר אומר ששחקן שלא שיחק תקופה, לא יכול לספק ניקוד רב? כמובן שלא! למשל:

- שחקן רכש חדש, אך איכותי. למשל, בעונה הנוכחית, שחקנים רבים הגיעו לפריימרליג. אין עליהם נתונים יבשים בתחילת העונה, אך אנו חייבים לתת אינדיקציה לכך שסביר להניח שכן יצליחו, אם הצליחו בליגות אחרות.

- שחקן איכותי שחזר מפציעה ארוכה.

- שחקן ספסל, אך שעתידי לקבל הזדמנות. לדוגמא, הקשר ההתקפי דייגו ז'וטה מליברפול לא סומן כשחקן הרכב. אך פציעות רבות של שחקנים מובילים שינוי את הכף לטובתו, והוא החל לצבור דקות משחק (ולצבור נקודות)

שיעורנו שקשה מאוד לקבל אינדיקציה עבור מקרים אלו – אך לשם כך, גייסנו את המדיה. אם נעבור על ציורים של ערוצים ומשתמשים נבחרים, שמפרסמים חדשות וידיעות ממסיבות עיתונאים, נוכל לקבל אינדיקציה חיובית או שלילית סביב שחקנים שלא היינו מקבלים במידע "היבש".

*הערת מקורות מידע עבור פרק זה: הידע המקצועי מעולמות הלמידה העמוקה ועיבוד השפה הטבעית מפרק זה מתבסס בעיקרו על הקורס "למידה עמוקה על מאיצים חישוביים"<sup>6</sup>. במקרים שבהם נלקח מידע ממקורות אחרים, מקורות המידע צוטטו וצורפו.*

<sup>6</sup> <https://vistalab-technion.github.io/cs236781/>

## AI Project - Fantasy Premier League Player

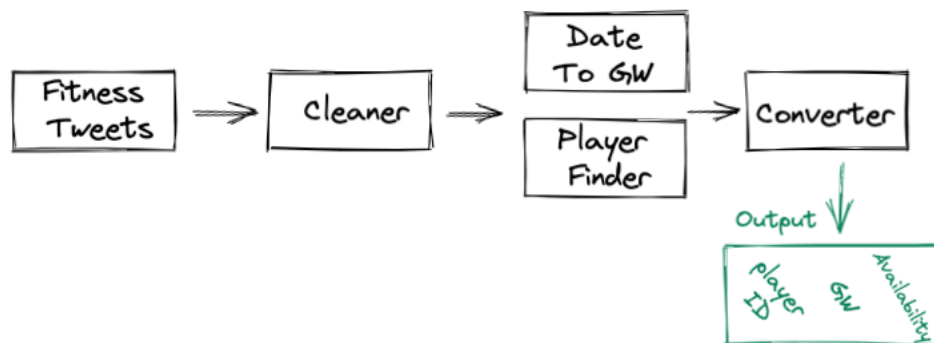
### מודל הכשירות

עבור מודל הכשירות, הציוצים הגיעו בצורה מאוד נוחה בעלת מבנה קבוע. המקור הינו מהימן, ופרסם את המידע ממסיבות העיתונאים הרשמיות ומההודעות הרשמיות של מועדוני הכדורגל. לפיכך, נדרשנו למימוש Parser בלבד על מנת לקבל את הפלט הרצוי.

### הפלט של מודל הכשירות:

Player id	fitness	GW
מזהה שחקן ייחודי	1 - fit 0 - not fit	המחזור שהפרדקציה רלוונטית עבורו

### הנדסת מודל הכשירות:



- הציוצים מה-DB הגיעו ללא עיבוד, ישירות מה-API של טוויטר.
- Cleaner מנקה את הציוצים ומסנן טקסט שאינו רלוונטי (בעזרת regex). בעיקר מנקים לינקים, תיוגים ומידע שנהוג להוסיף לציוצים. הקלט הוא המחרוזת של הציוץ, והפלט הוא רשימת מילים לפי מבנה קבוע:

תאריך	שם השחקן	קבוצה	סטטוס כשירות
תאריך פרסום הציוץ	שם השחקן	שם הקבוצה	סטטוס באחוזים, בקפיצות של 25%.

- שם השחקן תמיד הופיע בין התאריך, לבין מקף (" - "). ולכן, התאפשר לנו להגדיר בדיוק היכן שם השחקן מתחיל ונגמר (שמות השחקנים הם בני מילה אחת עד שלוש מילים, ולעיתים רחוקות אף יותר!).
- לעיתים מצוין תאריך החזרה לכשירות, אך כהשערה בלבד. לכן לא התייחסנו למידע זה.
- שם הקבוצה הופיע כתיג, ולכן המרנו כל תיג ל-`team_id` המוסכם.
- סטטוס הכשירות לרוב הינו 0% - לא כשיר, או 100% - כשיר. החלטנו להתעלם מסטטוס ביניים.
- Date to GW ממפה כל ציוץ למחזור. לדוגמא, אם ציוץ התפרסם בתאריך 10/12/2020, ומחזור 12 מתקיים ב-12/12/2020, הצמדנו את הציוץ ברלוונטי לקראת מחזור זה. לפיכך, הפרדקציה שלו כלפי הכשירות היא למחזור 12.
- Player Finder ממפה את שם השחקן ל-id שחקן (חד חד ערכי במערכת כולה). הוא משתמש בשם שהוציא ה-Cleaner וממפה אותו ל-id בעזרת האלגוריתם הפונטי.
- Converter מאחד את כל הנתונים, ומוציא את הפלט הנ"ל לפי כשירות בין 0 ל-1.

הערה: אתחול המודל יוצא מנקודת הנחה ששחקן כשיר, אלא אם נאמר אחרת. אם שחקן משנה מצב (מפצוע לכשיר למשל) המערכת מעדכנת זאת בהתאם. בתחילת העונה, מפורסמים עדכונים עבור כל הפצועים, כך שאנו יוצאים לדרך עם דאטה מהימן.

## AI Project - Fantasy Premier League Player

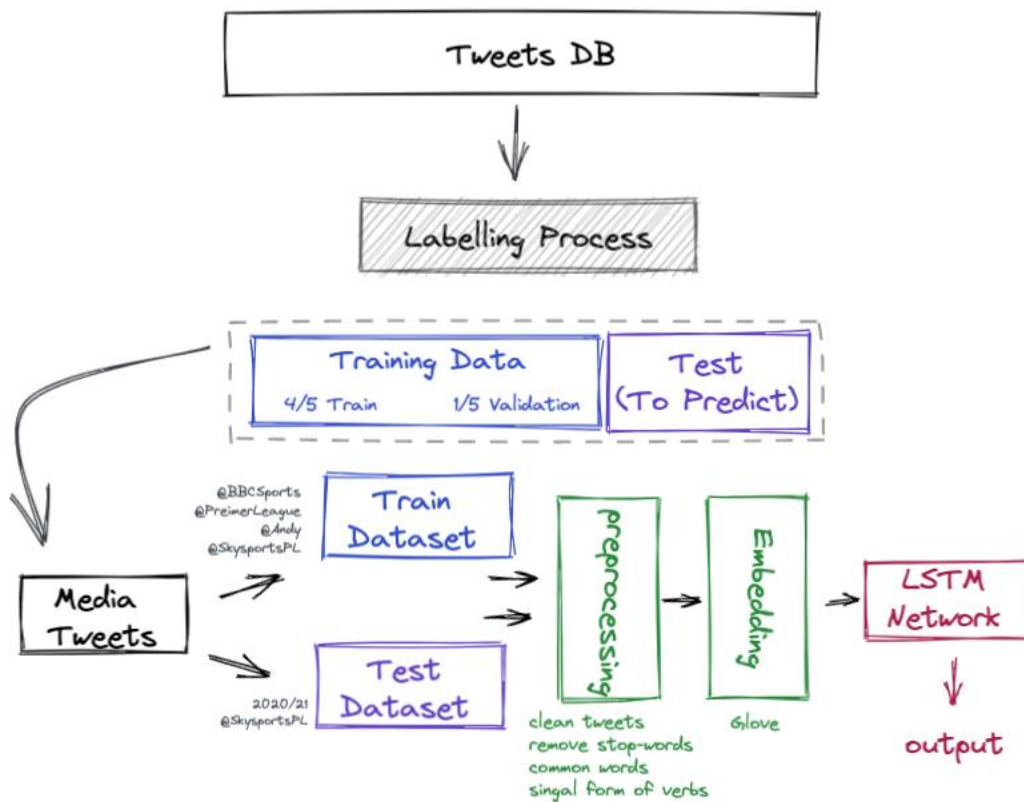
### מודל ההיפ

כפי שציינו בתחילת פרק זה, מטרת מודל ההיפ היא לתת מידע מזווית נוספת, שאינה מתקבלת מהנתונים הרשמיים. המודל מקבל כקלט ציוצים שפורסמו בין מחזורי המשחקים, ולאחר שמסנן ציוצים שאינם רלוונטיים, מעביר אותו במסווג שנבנה. התוצאה של המסווג היא סנטימנט שנאמר על השחקן

הפלט של מודל ההיפ:

Player id	hype	GW
מזהה שחקן ייחודי	$[-1,1]$ 1: negative sentiment 1: positive sentiment	המחזור שהפרדקציה רלוונטית עבורו

### הנדסת מודל ההיפ:



- איסוף הדאטה של הציוצים כפי שפורט בפרט על בניית מסד הנתונים.

### • Labelling Process – תהליך סינון תיוג הציוצים.

תהליך זה היה מורכב במיוחד. מחד גיסא, רצינו להשתמש בהמון ציוצים, אך התיוג שלהם מורכב ודורש התערבות אנושית. כמו כן, עלינו בכלל להחליט איזה מידע המסווג יוכל לקבל: אנו רוצים שהמסווג יפלוט, בתחום רציף, האם יש "באז" ו-"הייפ" חיובי או שלילי סביב השחקן. האם הוא הולך להצליח במחזור הקרוב, או שלא (יפשל, המאמן מאוכזב ממנו, ועוד).

- **סינון הציוצים:** השלב הראשוני בתהליך התיוג היה סינון כל הציוצים שאינם מהמבנה הבא:

**הציוץ יכול שם של שחקן אחד, שקיים במערכת, ואחד בלבד.**

## AI Project - Fantasy Premier League Player

הסיבה שהחלטנו לבצע זאת היא לשם הפשטות התחברית. ניתוח של משפט בעלת נושא מרכזי אחד (השחקן) פשוט הרבה יותר לניתוח הסנטימנט. אין בעיה להכין שמות עצם אחרים – כמו שם המאמן ושם קבוצה. לכן, בפסודו-קוד, הניפוי של הציוצים יבוצע כך:

1. אתחל  $i=1$  ורשימה ריקה של שמות שחקנים.
2. עבור על כל ציוץ, וסרוק בעזרת sliding-window בגודל  $i$  את המשפט.
  - a. עבור כל מחרוזת בגודל החלון הרץ, הרץ עליה את האלגוריתם הפונטי. במידה וקיבלנו id תקין – המחרוזת שללחנו הייתה של שחקן. שמור אותו ברשימת השמות.
  - b. אחרת, המשיך לסרוק.
3. אם  $i=4$ , צא מסריקת הציוץ. אחרת, הגדל את  $i$  ב-1, וחזור לשלב 2.

אנו לא יודעים את אורך השמות (מס' המילים בשם השחקן), או כמה שחקנים מופיעים בציוץ. לכן, אנו סורקים כל רצף של מילים (עד 4 מילים) בעזרת חלון רץ, ומזהים שמות. כעת, כל ציוץ שאינו עומד בתנאי – שם אחד, ואחד בלבד – נזרק מהמאגר. גם כאן, בדומה למודל הכשירות, ביצענו התאמה בין תאריך פרסום הציוץ לבין מחזור המשחק.

לאחר שלב זה נשארו עם קצת יותר מ- 5,500 ציוצים לאימון.

○ **תיוג הציוצים:** השלב הבא היה תיוג הציוצים עבור שלב האימון. רצינו לתייג את כל הציוצים באופן אנושי, להיות מנותקים לגמרי ממודל למידה אחר – אך המאגר לתיוג היה גדול מידי. לכן, עבדנו בשיטה "משולבת" בין מודל קיים לבין תיוג אנושי ע"י מספר אנשים.

### תהליך התיוג המשולב התבצע בצורה הבאה:

1. תיוג כלל הציוצים ע"י TextBlob. ספרייה זו מאפשרת להוציא סנטימנט עבור משפט (בין 1 ל-1-). זוהי ספרייה מפורסמת ומתוחזקת, שאחוזי הדיוק שלהם גבוהים מאוד.
2. סינון הציוצים שאינם חיוביים או שליליים באופן מובהק (מעל 0.1 או מתחת 0.1- בהתאמה). הציוצים שבין תחומים אלו מוגדרים ניטרלים.
3. נתנו לקבוצת אנשים לתייג 1,200 מהציוצים הללו באופי ידני. נאמר להם לתייג האם הציוץ חיובי או שלילי. חלק מהציוצים שתיוגו ע"י האנשים היו חופפים, לשם בקרה על כנות המשיבים. 4 מהמשיבים מכירים את הפנטזי ו"נושמים" את עולם הכדורגל, ו-2 משיבים אינם מכירים את הפנטזי, אך קצת מכירים את עולם המושגים. נאמר להם לתייג במקבצים קצרים, של 10-20 ציוצים, כדי להימנע מעייפות ומסימון שווא. בסופו של דבר, כל אחד מהנשאלים תייג 250 ציוצים באופן ידני – 200 מהם יחודיים, ו-50 מהם חופפים לצורך בקרה.

--- **וכאן תודה מיוחדת לעדיאל נחום, ברק מרום, אליעד צמח, אייל כנפי, דוד אוחנה והילית נגר על**

**התיוג הידני. מסירותכם והשקעתכם תיזכר לדורות! ---**

- ראינו התאמה כמעט מושלמת בין הציוצים שתיוגו כחיוביים ע"י TextBlob לבין אלו שתיוגו ע"י חברינו היקרים. 19 ציוצים בלבד קיבלו תיוג הפוך. זו טעות שאנו יכולים "לספוג" עבור דאטה גדול יותר, ובכל זאת ולצורך השלמות ההסבר שהגענו אליו בגלל השגיאה ציוצים אלו היו נוטים באופן יחסי לניטרלים, או שתיוגו בטעות אנושית.
- כל הציוצים שהיו חופפים לצורך הבקרה קיבלו תשובה זהה ע"י כל הנשאלים!

## AI Project - Fantasy Premier League Player

### • חלוקת הדאטה לאימון: Training & Validation Set

השתמשנו בשיטת cross-validation לטובת אימון וכיוון הפרמטרים של המודל. כלומר, יציאת מודל מ-80% מהדאטה לאימון, יצירת מודל מסווג, ואז בדיקתו ע"י 20% מהדאטה הנותר. כך נתנו ציונים למודלים השונים שבחנו, עם הפרמטרים השונים. המודל שקיבל את הציון הגבוה ביותר – הוא המודל שאיתו השתמשנו עבור הוצאת הפלט הרשמי עבור אלגוריתם השחקן. התהליכים הבאים שנתאר רלוונטים עבור קבוצת האימון וגם קבוצת הפרדיקציה.

### • עיבוד מקדים – Preprocessing

השלב שבוא עובדו הציוצים לקראת הכנסתם למודל. חלק מהשלב הזה נבדק וכוון (פירוט בשלב הניסויים). בפועל, עבור המודל הסופי, התהליך שבוצע:

- ניקוי הציוצים מביטויי טוויטר (כמו #, @, rt עבור retweet וכדומה), קישורים, תמונות. Lower-case לכל הציוצים.

- הורדת stop-words [נבחן בניסוי]

Stop-words – הכוונה למילים הנפוצות תחבירית בשפה שעליה עובדים (במקרה שלנו, אנגלית). מילים אלו מורדות, במחשבה שאינם מוסיפות אינפורמציה אלא רק מכבידים על המודל. לדוגמא: "a", "is", "are", "the". המחשבה הייתה שמכיוון שאנו משתמשים בנושא מרכזי אחד במשפט – השחקן – המשמעות שלהם תהיה פחותה. אך הדבר נבחן בניסוי והופתענו לגלות כי המודל הצליח יותר ללא הורדת stops words.

- נרמול הפעלים: "כבש", "כובש", "יכבוש" (במשמעות של לכבוש שער) בעלי משמעות סמנטית דומה. נרמלנו את הפעלים לזמן יחיד כדי להקל על המודל.

### • Embedding

Word-embedding משמר את משמעות המילים במרחב, והופך את המילים ומשמעותם הסמנטית לוקטורים בעלי ערך נומרי שאיתם הרשת תוכל לעבוד. כלומר, אנו לוקחים מילה, והופכים אתה לוקטור. מודלים אלו, כמו Word2Vec שבו השתמשנו, יוצרים מרחב וקטור אדיר (של מאות מימדים), ומילים החולקות קשרים משותפים יותר, בעלי מרחק קטן יותר אחד מהשני.

נדגים: למילים "שער", "יוצא מהכלל", "ניצחון" מקושרים כמובן להצלחה בכדורגל. בעוד בן אנוש יכול להסיק זאת בקלות, למכונה הדבר מורכב יותר. לשם כך צריך מודל שידע לבצע זאת בצורה חכמה – אך הדבר דורש זמן ומשאבים רבים.

נבחן את שכיחות אוסף המילים הבא, שלקוח מתוך כתבות (דמיוניות) מענפי הספורט הבאים:

GW	כדורגל	כדורסל
כדור	100	90
שחקן	120	120
מערך	90	2
חמישייה	0	70
פסק-זמן	5	100
שוער	120	0

קל לראות שיש מילים שנמצאות במרחב משותף, כמו כדור, אך יש גם מילים שנמצאות רק במרחב של אחד מהענפים.

## AI Project - Fantasy Premier League Player

קורפוס מילים הוא אוסף בנושא מסוים. היינו שמחים להשתמש בקורפוס בנושא כדורגל אנגלי, אך נסתפק בקורפוס של ציורים מטוויטר - GloVe<sup>7</sup>

לשם תהליך זה נעזרנו בתהליך של Word2Vec, אשר מבצע טרנספורמציה: ממילים, לקטורים. הדבר נעשה באמצעות מודל מאומן. המודל אשר הורדנו מGloVe מכיל ייצוגים וקטורים בגדלים של 50-200 מימדים שבהם יכלנו להשתמש, ומכילי מיליוני מילים. כל מילה כזו ברשימה, נקראת Token. כך יכלנו להצמיד למילים מהציוצים וקטורים. הטוקנים קריטים, ולמעשה ממירים מילים למספרים שאיתם הרשת הנוירונית תוכל לעבוד.

לדוגמא, נוכל לבדוק וקטור של מילה מסוימת –

```
nlp_model['goal']
```

```
([-0.5415519, 1.5530779, -0.31864733, 2.1040018, 2.4623108, 0.1254964, 0.14440827, -1.022954, -0.49929366, 1.2939268, ],)
```

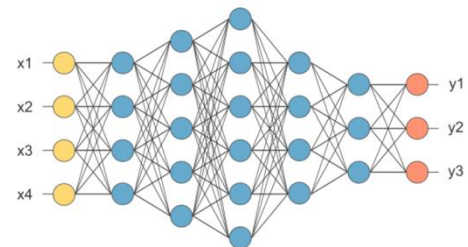
כאן ניתן לראות וקטור במרחב של 10 מימדים. בפועל, השתמשנו ב-200 מימדים. עבור מילים שלא הופיעו במרחב, הגדרנו וקטור מאופס. הסיכוי שמילה לא תופיע במאגר אפסי ביותר – פרט, אולי, לשמות אזוטריים של חלק מהשחקנים או המאמנים שמדברים עליהם ומופיעים בציטוט. נציין שהדבר לא היווה בעיה עבור שמות השחקנים או הקבוצות: אותם אנו מחלצים טרם הכנסת הציץ לרשת העמוקה. כלומר, מבחינתנו, העיקר זה "שאר המשפט" ואכן התוצאות היו טובות דיים בהקשר זה. לבסוף שלב זה, אנו "ניישר קו" בין כל הוקטורים, נרפד את הוקטורים הקצרים כדי שיהיה להם אורך אחיד והם יהיו מוכנים לרשת הנוירונית.

**למעשה, שכבת ה- embedding היא השכבה הראשונה ברשת העמוקה שלנו.**

- **הרשת העמוקה שלנו, ומבוא קצרצר ללמידה עמוקה**  
הרשת שבה בחרנו להשתמש היא LSTM, אך לפני שנפרט עליה, נסקור בקצרה את עולם המושגים של הלמידה העמוקה שבו השתמשנו:  
- **רשת נוירונים:** מודל חישובי, שיחידת הבסיס שלו מכונה נוירון. רשת זו מכילה מספר משתנה של שכבות הקשורות זו בזו. כל רכיב ברשת מקבל קלט, מבצע פעולה ומעביר את הפלט הלאה. שכבת הקלט היא השכבה הראשונה, הפלט היא האחרונה ושכבות הביניים הם השכבות "הנסתרות" (Hidden Layers). הפעולות על כל נוירון ממושקלות, ועיקר תהליך הלמידה הוא לעדכן את המשקולות על כל רכיב. **תהליך הלמידה מתבצע ע"י קבלת קלט, הוצאת פלט כסיווג/רגרסיה וחישוב loss עבור התוצאה.**

fully connected layer - FC. כל רכיב תלוי בשכבה הקודמת. רשת עמוקה בסיסית, בה החישוב עבור כל רכיב  $y$  מבוצע בשיטה הבאה:

$$y_i = \phi(W_i y_{i-1} + b_i), W_i \in \mathbb{R}_{n_i \times n_{i-1}}, b_i \in \mathbb{R}_{n_i}$$



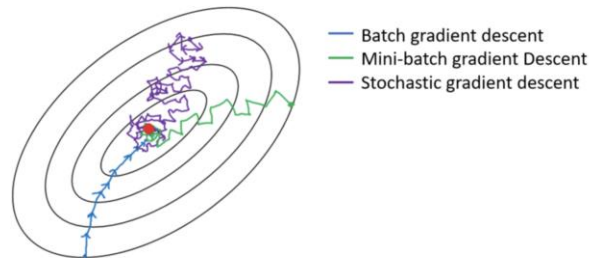
- **פונקציית אופטימיזציה – backpropagation.**

<sup>7</sup> <https://nlp.stanford.edu/projects/glove/>



## AI Project - Fantasy Premier League Player

מטרתה למזער את פונקציית ה-loss. קיבלנו פלט מהרשת, וחישבנו ערך loss, וכעת אנו רוצים לבצע "תיקונים ברשת" ע"י חזרה אחורנית. הפונקציה המוכרת והפשוטה, באופן יחסי היא Gradient Descent. אנו מחשבים את הכיוון של הגרדיאנט מהרכיב שעליו אנו נמצאים ו"מתקדמים" לכיוון הנגדי לגרדיאנט. כך, למעשה, אנו מתקדמים למזעור הפונקציה.



אילוסטרציה של סוגים שונים של Gradient Descent<sup>8</sup>

השיטה הקלאסית פחות שימושית, וישנם שיטות שונות. למשל, SGD (סטוכסטי) אנו מבצעים את החישוב ע"פ הדוגמא הנוכחית, ויוצרים תנודתיות גדולה יותר אך פרקטית יותר. בחירת פונקציה נכונה והפעלת המתודות הטובות ביותר קריטית לזמן האימון ולנכונותו. למשל, לא נרצה לעצור ב"אובפים" או מינימום מקומי בפונקציה. לשם כך מוסיפים מומנטום לפונקציה: עקרון שבו מבצעים ממוצע עבור היסטוריית הערכים ובעזרתם לעדכן את המשקלים. אחת מהמתודות המוכרות המיישמת את עקרון זה, והמוצלחות לרוב, הינה Adam- הפונקציה שבה השתמשנו עבור ה-optimizer שלנו.

- בעיות גרדיאנט: Exploding / Vanishing.

הבעיות העיקריות בהקשר הגרדיאנט והחישוב שלו (המהותי לעדכון המשקולות ברכיבים) הם מצבים בהם הגרדיאנט "מתפוצץ" או "נעלם". כלומר הערכים שלו גבוהים או נמוכים מידי, והדבר פוגע במודל. ישנם מספר טכניקות לטיפול בבעיה זו: למחוק קשרים בין רכיבים, לדלג על רכיבים מסוימים. ישנם טכניקות נוספות ועל אחת מהם נפרט ממש בקרוב.

RNN

<sup>8</sup> <https://suniljangirblog.wordpress.com/2018/12/13/variants-of-gradient-descent/>

## AI Project - Fantasy Premier League Player

רשת המשתמשת ללמידה ממידע סדרתי – למשל, אוסף של מילים המופיעות אחת אחרי השנייה (משפט). היא כוללת "רכיב זיכרון". רשת זו יכולה להכיל גם "חזרה אחרונית", בניגוד לרשתות הקלאסיות. עקב יכולות "שימור הזיכרון" שלה, רשת זו משתמשת רבות לתחום עיבוד השפה הטבעית – מניתוח סנטימנט, ניתוח אובייקטיביות ועד ליצירת משפטים באופן עצמאי<sup>9</sup>.

רכיב בודד ברשת RNN. כאשר:

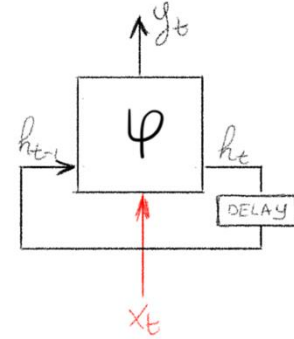
$x_t$  – current sample

$h_{t-1}$  – last state

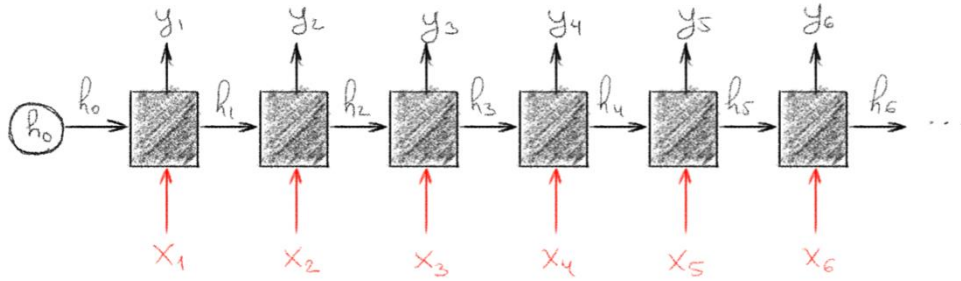
$y_t$  – current output

$h_t$  – current state

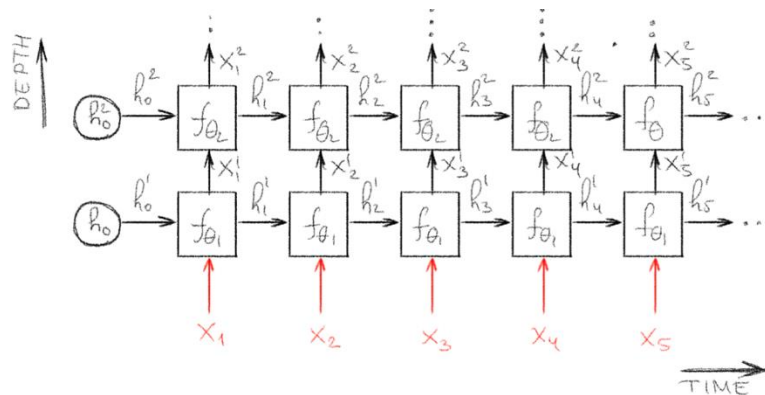
כלומר, קיימת "העברה קדימה" וגם "זיכרון", לכל רכיב בנפרד ברשת.



ובאשר הרכיבים מחוברים בצורה סדרתית, הרשת מסתדרת כך:



שכל רכיב הוא אותו הרכיב שהוצג לעיל. בנוסף, רשתות אלו יכולות "לחבור יחדיו" לרשת עמוקה של RNNים:

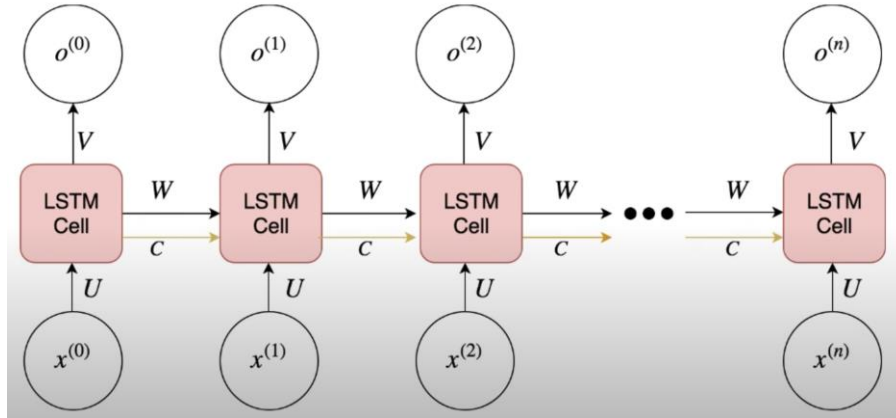


<sup>9</sup> © אמירה זו מתבססת על הקורס "למידה עמוקה על מאיצים חישוביים". בנוסף, האיורים בחלק זה מובאים ממחברות התרגול של קורס זה

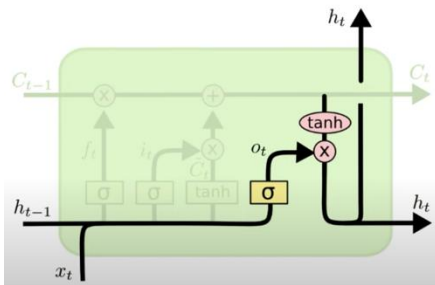
# AI Project - Fantasy Premier League Player

## הרשת שבחרנו - LSTM

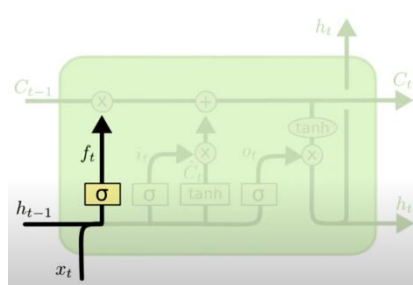
**LSTM** הינה רשת RNN, אשר גם לה יכולת זיכרון, אך יכולת "לשכוח" מידע באופן מודע. כלומר, יחידות הבסיס של הרשת מכילות שערים של קלט, פלט וגם שכחה. השימוש בשערים אלו יכולים לאפשר לרשת הבנת הקשר (Context) מה שהופך אותה לחזקה בהקשרי שפה. השימוש ברצפים וביכולת זיכרון נותנת יתרון גדול למודל זה לעומת מסווגים שעובדים "מילה – מילה" ללא הקשר. הרשת מכילה תאי LSTM בצורה הבאה:



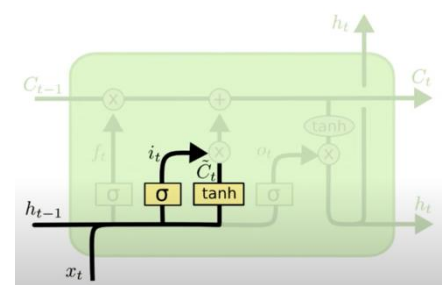
כאשר כל רכיב זה, מכיל 3 שערים:



שער פלט, האם הפלט נראה

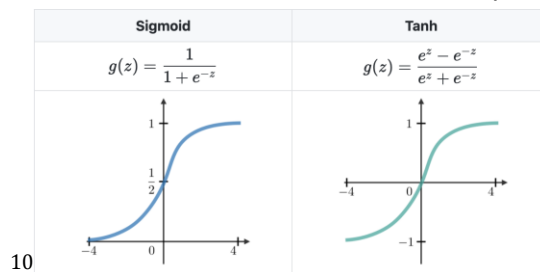


שער שיכחה, מאפס את התא



שער הכניסה, האם התא מעודכן

כך המידע זורם, או שלא, בין התאים ברשת. ניתן לראות שבשערים מופעלות פונקציות אקטיבציה: sigmoid בכלום, ו-tanh בחלקם. אלו עוזרות, בין היתר, לפתור את בעיות הגרדיאנט שצינו קודם, ומכריעות את צורת הפלט לרכיב הבא.



פרט למודל המרכזי, הרשת יכולה להכיל את השכבות הבאות:

<sup>10</sup> <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks#architecture>

## AI Project - Fantasy Premier League Player

- שכבת dropout שכבה ברשת אשר מאפסת באופן אקראי חלק מהמשקלים על הרכיבים ברשת. הדבר עוזר להתמודד עם overfitting באימון. וגם לכאן, יש שיטות שונות כמו recurrent dropout, שמפעיל את הדרופ בין נקודות זמן שונות בשכבות.
- שכבת Pooling הינה שכבה שבנה מבצעים חישוב מסוים – למשל, ממוצע – על חלק מהנתונים, וממפים אותם למרחב קטן יותר.
- שכבת Dense הינה שכבה אשר מסווגת את הטקסטים לתחום הרציף שלנו. תהיה השכבה האחרונה ותמפה את התוצאות לסנטימנט חיובי או שלילי.

בחרנו להשתמש ב-LSTM במודל הסופי שלנו **[נבחן בניסוי]**. פירוט השכבות בחלק של הניסויים.

## AI Project - Fantasy Premier League Player

לאחר שיש לנו את חיזוי הנקודות, הכשירות והייפ עבור כל שחקן, בכל מחזור, הגיע הזמן לממש את האלגוריתם המורכב שמבצע את הפעולות הנדרשות בכל מחזור.

נזכיר, כי הקלט של האלגוריתם, עבור כל מחזור משחק, הוא רשומות מהצורה הבאה עבור כל שחקן:

מזהה שחקן	קבוצה	xP			כשירות	הייפ
		1	2	3		
מזהה יעודי לכל שחקן.	שם הקבוצה שבה השחקן משחק	צפי למחזור הקרוב, המחזור שאחריו ושניים שאחריו.	ערך בינארי עבור כשיר או לא כשיר.	ערך רציף בין 1 ל-1- עבור האם קיים סביב השחקן הייפ בתקשורת.		

מרחב האפשרויות עצום – גם לאחר שאנו מקבלים פרדיקציות עבור שחקנים, ונוכל לצמצם את מרחב האפשרויות שלנו לתת קבוצה של השחקנים עם הניקוד הבולט בכל עמדה – עדיין, תכנון אסטרטגיה עבור חילופים וצ'יפים היא משימה מורכבת. אתחול ההרכב (בעיית Knapsack) היא בעיה NPC ובמשחק שלנו יש קושי נוסף – הגבלות על תקציב, על העמדות וכן על מס' שחקנים מכל קבוצה.

**לשם כן החלטנו לממש אלגוריתם מקבילי שמחפש במרחב הסגלים האפשריים את הסגל עם ניקוד גבוה ככל שניתן. היוריסטיקה היא הקלט (הטבלה לעיל), והחיפוש נעשה ע"י חיפוש המדרון התלול ביותר.**

כחלק מהאלגוריתם, הוכנסו פרמטרים הניתנים לשינוי כדי שנוכל לבדוק הרצות שונות ולנסות לקבל תוצאות טובות ככל הניתן. הפעם לא מדובר על מודל מאומן, אלא על הרצת האלגוריתם בצורה בלתי תלויה, כדי לבחון את התוצאות השונות שלנו.

הפרמטרים שנבחנו [יפורט בחלק הניסויים] –

- Threshold על חילוף וחילוף כפול.
- משקולות על חשיבות ההייפ.
- משקולות על תחזית הנקודות לפי שבועות קדימה.
- פרמטר כמות הקבוצות שמריצים באתחול ראשוני (ומשם מנסים להגיע למקסימום) .

### פאסדו קוד והסבר כללי על האלגוריתם

אלגוריתם בחירת השחקנים מתחלק לשני חלקים:

- אתחול קבוצה חדשה
- הרצה של מחזור בודד

1. בעיית אתחול קבוצה חדשה היא בעיית תרמיל הגב עם כמה הגבלות נוספות:

- מחיר שחקן שקול למשקל בפריט והמשקל הכולל המותר הוא 100.
- המשחק מחלק את השחקנים לעמדות ואנו יודעים בדיוק כמה שחקנים יש לקחת מכל עמדה.
- הגבלת הסכום היא כללית ואין חלוקה לעמדות.
- מותר לבחור לכל היותר 3 שחקנים מאותה קבוצה.

מבאן, הטרמינולוגיה היא **עוצמה – של שחקן או קבוצה**. מדובר על הניקוד החיזוי בשקולל פרמטרים נוספים (כשירות והייפ). בחירת שחקנים לפי "הטוב ביותר", כלומר, **בעל העוצמה הגבוהה ביותר**.

בעיה זו היא בעיה NPC ולכן הדרך שבה בחרנו לפתור אותה היא בצורה איטרטיבית:

## AI Project - Fantasy Premier League Player

1. האלגוריתם מקבל את התחזית לביצועי במחזור הקרוב.
  2. מתוך השחקנים, הוא בוחר בצורה אקראית 15 שחקנים לפי החוקים ללא הגבלת משקל ומחליף הצורה אקראית שחקנים לשחקנים זולים יותר שלא יפגעו בחוקיות הקבוצה עד שתקציב הקבוצה עומד בדרישה ההתחלתית.
  3. האלגוריתם עובר על הקבוצה ומחפש בכל פעם שני חילופים שישפרו את הקבוצה ויעמדו בדרישות. שלב זה לוקח את הקבוצה האקראית שלקחנו ומחפש מקסימום מקומי על פי התחזיות של המודל. חילוף בודד אינו משיג חיפוש מקסימום במרחב גדול מספיק לדוגמה אם הוצאתי מגן אוכל להחליפו רק במגן לעומת שני חילופים שניתן להוציא שילוב של עמדות וקבוצות רחב יותר. ביצוע 3 חילופים במקביל לא העלה את התוצאות בצורה ניכרת אך דרש זמן חישוב גדול בצורה משמעותית ולכן חיפוש המקסימום מבוצע בשני חילופים.
  4. כל בחירה כזו של קבוצה אקראית וחיפוש מקסימום מקומי ישיג ממנה הוא בלתי תלוי בחישובים אחרים ולכן באתחול שחקן חדש, על מנת להגדיל את הסיכוי למצוא מקסימום מקומי מקסימאלי, אנו מאתחלים בשלב זה מספר קבוצות אשר מחושבות **בצורה מקבילית**. ניתן להכניס בפרמטר את מספר הקבוצות שרוצים לבדוק.
  5. בסיום הריצות נבחרת קבוצה אחת שהתחזית ניקוד שלה הכי טובה.
- 
2. בהרצה של **מחזור בודד** יבוצעו השלבים הבאים:
    1. נבדוק מה החילוף הבודד הטוב ביותר שאנו יכולים לבצע.
    2. אם יש לנו שני חילופים נבדוק מה הם שני החילופים הטובים ביותר שנוכל לבצע.
    3. נבדוק את השיפור החזוי שנשיג על אחד מהציפים שנשארו לנו:
      - 3.1 Wild\_card: שלב זה שקול לאתחול קבוצה חדשה והשוואתה לקבוצה שיש לנו לאחת שביצענו חילוף או שניים בהתאם למה שיכולנו, אם השיפור גדול מפרמטר קבוע נפעיל ציפ זה.
      - 3.2 Free\_hit: גם כאן יבוצע אותו תהליך כמו Wild\_card עבור פרמטר שונה.
      - 3.3 Bench\_bust: אם לא הופעלו ציפים אחרים נבדוק האם הספסל צפוי לתת ניקוד לפחות כמו פרמטר מתאים ואם כן נפעיל אותו.
      - 3.4 Triple\_captan: הקפטן נבחר בצורה פשוטה – השחקן שצפוי לתת את מספר הנקודות הגדול ביותר השבוע. אם הקפטן צפוי לתת ניקוד גבוהה לפי פרמטר שנקבע ולא הופעל ציפ אחר נפעיל אותו.
  3. היררכיית הפעלת הציפים היא לפי הסבירות שיופעל כל צ'יפ.
    - 3.5. אם נותרו לנו ציפים כמספר המחזורים או Wild\_card במחזור ה-19 שלאחריו הוא נמחק, נפעיל את ציפ הראשון בהיררכיה שנותר ערך חיובי.
    4. אם לא הופעלו הציפים wild\_card או Free\_hit, נבדוק אם יש לנו שני חילופים אם לבצע את שניהם ייתן ניקוד מספיק גבוהה נבצע שני חילופים.
    5. אחרת נבצע חילוף בודד ונעביר את החילוף הנותר למחזור הבא.
    6. אם לא היה חילוף נוסף ולא הופעלו שני הציפים נבדוק האם חילוף בודד ייתן מספיק ניקוד, אם כן נבצע אותו אם לא נשמור אותו למחזור הבא.
  3. בחירת הרכב:

## AI Project - Fantasy Premier League Player

3.1. בוחרים מתוך הסגל בכל עמדה את השחקנים עם הביצועים הצפויים הכי טובים – **כלומר**,

**העוצמה הכי גבוהה** - לפי המינימום הנדרש בעמדה: שוער, 3 בלמים, 3 קשרים וחלוץ

3.2. יתר השחקנים יכנסו להרכב לפי הביצועים שלהם כמובן ללא השוער שם משחק בדיוק 1.

3.3. כל היתר יהיו על הספסל.

4. הערכת עצמה של קבוצה ושחקן:

4.1. הערכת העוצמה של הקבוצה היא סכום העצמות של שחקני הסגל.

4.2. **הערכת העוצמה של שחקן נקבעת לפי הניקוד החזוי לו בשלושת המשחקים הקרובים, האם**

**הוא צפוי לשחק – כשיר - במחזור הקרוב, ומה ההיפ סביבו ברשתות החברתיות.**

4.3. שקלול הפרמטרים מושפע מהאם השחקן בהרכב שם תנתן עצמה בעיקר לפי תחזית למחזור

קרוב לעומת ספסל שם לא ינתן ניקוד על מחזור קרוב ולכן לא נתחשב בו כמו גם בתחזית אם ישחק או לא.

### עבודה מול סימולטור המשחק

סימולטור המשחק מקבל כקלט את ההרכב שבחר אלגוריתם השחקן. הוא מוודא כמובן שההרכב חוקי והפעולה שבוצעה חוקית, ומנקד לפי הציונים בפועל את השחקנים.

סימולטור המשחק רושם ללוגים את הפלט של הפעולות בוצעו, תקציב וההרכב הנוכחי.

## AI Project - Fantasy Premier League Player

### 4.6 אלגוריתמים ושיטות נוספות במערכת

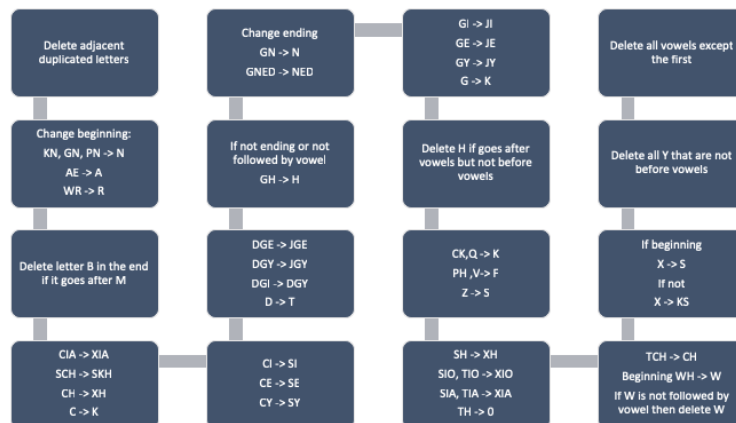
#### 4.6.1 כתיב פונטי

בעת עיבוד המידע הגולמי עבור יצירת Dataset, וכן בעיבוד הציוצים שהתקבלו מה-Twitter, נתקלנו בבעיה קטנונית אך קריטית: במקורות מידע שונים, ולעיתים גם באותו מקור, שמות השחקנים נכתבו בצורות שונות ונשמרו בקידודים שונים. על מנת שהרכיבים השונים במערכת ידברו "באותה שפה", היינו צריכים למצוא דרך לתרגם את הנ"ל לקוד חד-חד-ערכי המקושר לשחקן.

לדוגמה: אלכסיס סנצ'ז (Alexis Sánchez), מלך השערים של נבחרת צ'ילה, הופיע פעם אחת כ-Alexis Sñfñnchez, פעם אחרת כ-Alexis Sñfññnchez, ופעם נוספת כ-Alexis Sánchez. דוגמה נוספת היא החלוץ הבלגי רומלו לוקאקו, אשר הופיע גם כ-Romelu Lukaku וגם כ-Romelu Lukakku.

המרת השמות לקידוד אחיד (בגון ASCII / Unicode / UTF-8 / ISO-8859-1) אינה רלוונטית במקרה של ידידינו הבלגי, ויצרה מגדל בבל חדש עבור הצ'יליאני. הנחנו שאנחנו לא הראשונים שנתקלו בבעיה כזו, ולאחר חיפושים וניסויים הגענו לארץ המובטחת – Double Metaphone Algorithm. אלגוריתם זה, שפורסם ע"י Lawrence Philips בשנת 2000, הוא גלגול של Soundex Algorithm שפותח בשנת 1918 כדי להתמודד עם... גלי הגירה מאסיביים לארה"ב! לא מעט סנצ'זים, ביורנים (Björn) וחבריהם עשו דרכם לארץ האפשרויות הבלתי מוגבלות, והעמידו לא מעט קשיים בפני הפקידים שניסו להבין איך לבטא ו/או לכתוב את שמם ואת שם העיר ממנה הגיעו. על מנת להתגבר על הבעיה, Robert C. Russell פיתח את הרעיון הפשוט אך גאוני הזה.

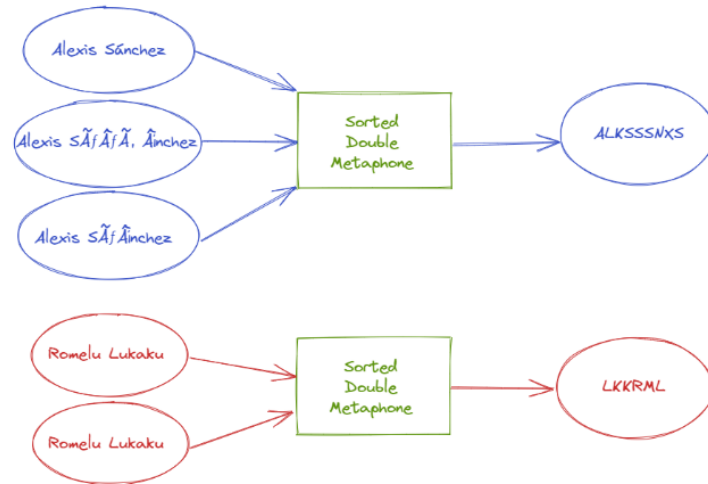
דיאגרמה כללית המתארת את אופן עבודת Metaphone Algorithm (לקוח מהאתר [medium.com](https://medium.com)):



האלגוריתם מייצג רצפי תווים באופן עקבי ומתמודד היטב עם תווים לא-לטיניים, ולאחר ביצוע טוויק קטן נוסף בדמות מיון שמות שמות השחקנים (עבור התמודדות עם סדרים לא קבועים של שם פרטי -> אמצעי -> משפחה), קיבלנו מעל ל-95% דיוק בהמרת שמות השחקנים לכדי ייצוג פונטי חד ערכי.



## AI Project - Fantasy Premier League Player



ראוי לציין כי קיימת גרסה מתקדמת יותר, Metaphone3, אשר על פניו מתאימה יותר לצרכינו – אך היא אינה מצויה לשימוש חופשי, שלא כמו Double metaphone אשר זמינה באמצעות הספרייה [Metaphone](#).

### 5. ניסויים, כיוון הפרמטרים ותוצאות

#### 5.1 הצגה כללית של תוכנית הניסויים

לאחר תכנון ובנייה של כל אחד מריבבי המערכת, שרטטנו את תוכנית הניסויים. באופן כללי, אין גבול לכמות הפרמטרים שנוכל לבדוק. בעיקר את מודלי הלמידה העמוקה, שכיוון הפרמטרים שם הוא תורה בפני עצמה – ולרוב, ללא יכולת אמיתית להבין מדוע פרמטר מסוים הצליח ומה פחות. לפיכך, ניסנו להתמקד על מודלים ושיטות שהוכחו כיעילות (ממאמרים ומקורס מאיצים חישוביים) ואותם ניסנו לכוון וללמוד.

עקרונות (ברורים מאליו, אך שהיה חשוב להדגיש) עבור כלל הניסויים:

- הניסויים של מודל חיזוי הנקודות נעשו אך ורק על עונות קודמות.
- הניסויים של מודל ההיפ נעשו אך ורק על קבוצת הולדיציה. כלומר, לא על ציורים שעליהם הוצאנו את הפרדקציה.
- עבור אלגוריתם השחקן, העבודה מול הפרמטרים היא בצורה שונה: לא מדובר על מודל מאומן מתחום הבינה המלאכותית, אלא על אלגוריתם איטרטיבי בפותר בעיית חיפוש מורכבת במרחב הקבוצות. הוא ניתן לכוון ע"פ פרמטרים, ואכן בדקנו פרמטרים שונים איתם סימלצנו את עונת המשחק.

## AI Project - Fantasy Premier League Player

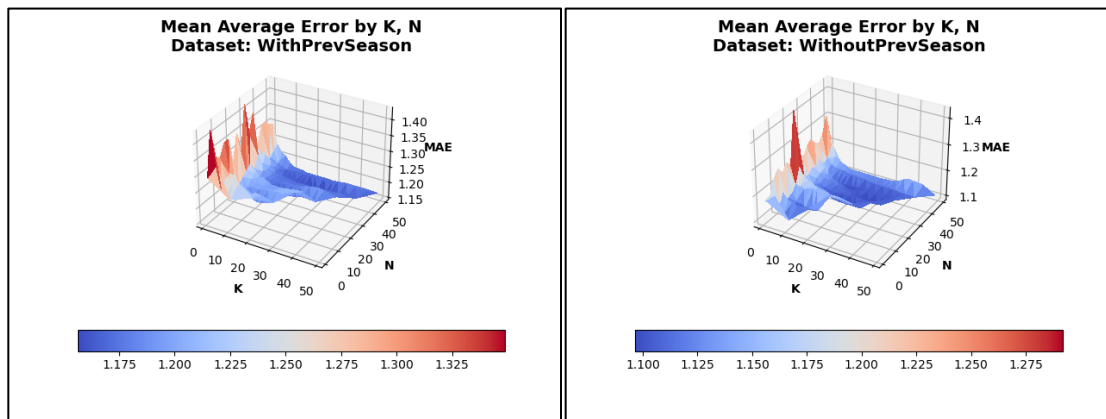
### 5.2 ניסויים וכיוון מודל חיזוי הנקודות

כפי שהוסבר לעיל, נקטנו בשתי גישות מרכזיות: סטאטית ודינאמית. כל גישה דרשה מספר ניסויים ובדיקות, בין אם של Hyper parameters ובין אם של ארכיטקטורה כללית עבור המודל. לכל אורך חלק זה, ועל מנת להפחית אי התאמות, החלטנו להשתמש במטריקת Mean Average Error עבור מדידת טיב המודל, שכן זוהי ההסתכלות הכי טבעית (בעינינו) עבור שגיאה במימד יחיד, וכן היא משקפת בצורה הישירה ביותר את השפעת שגיאות הרגרסיה על תוצאות המשחק. נרחיב כעת על הניסויים המשמעותיים שבוצעו עבור כל גישה:

#### 5.2.1 גישה סטאטית

ראשית, נציין כי גישה זו הובילה לתוצאות לא רעות, אך השערנו הייתה כי השיטה הדינאמית תוביל לתוצאות מוצלחות יותר. ואכן כך היה, אך הדבר הוכרע כמובן בסדרה של ניסויים. עם זאת, גישה והמודלים שבחנו פה עזרו מאוד לפיתוח המודל הסופי.

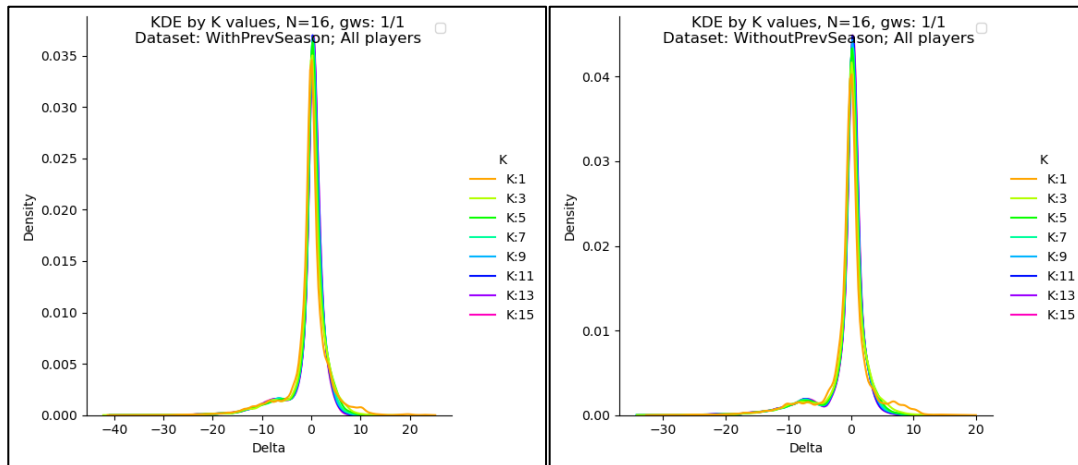
לאחר שביצענו מספר ניסויים לא מאוד מעניינים עם עצי רגרסיה פשוטים ו-KNNים לא מתוחכמים על מנת לקבל מושג כללי על התנהגות הדאטה, השלב הבא היה לעבור ל-**Random Forest**. חילקנו את ה-Train set ל-N קבוצות, כך שכל קבוצה תכיל מידע על  $\frac{|Dataset|}{N}$  מופעים (שחקן + מחזור) ועבור ערכי K משתנים חזינו לכל מופע ב-Validation set את ממוצע הניקוד שקיבל מ-K העצים הקרובים אליו ביותר, כפי שהוסבר בפרקים הקודמים. נציג תחילה את השגיאות הממוצעות שהתקבלו עבור כלל המופעים, בחלוקה לפי ה-Dataset שלם שלנו עבור ערכים של  $N \in [1, 49]$ ,  $K \in [1, N]$ . ניתן לראות כי באופן כללי השגיאה הממוצעת מפתיעה לטובה – התייצבות על סביבות ה- $MAE \approx 1.17$ . ככלל אצבע ערכי K, N גבוהים יותר הניבו תוצאות טובות יותר – עם הסתייגות עבור WithoutPrevSeason אשר נראה כי החל מ- $N, K \geq 40$  החל במגמה הפוכה.



על מנת לקבל תמונת מצב טובה יותר ומעמיקה יותר על התפלגות השגיאות, החלטנו לייצא את ה-Kernel Density Estimation שלהן, תוך הסתכלות על כלל המופעים לעומת אלו שבהן ניקוד השחקן היה שונה מאפס. כדי להקל על הקריאות, ייצאנו לכל ערך N גרף עצמאי המכיל את ששת ערכי ה-K שהובילו לערכי ה-MAE הנמוכים ביותר.

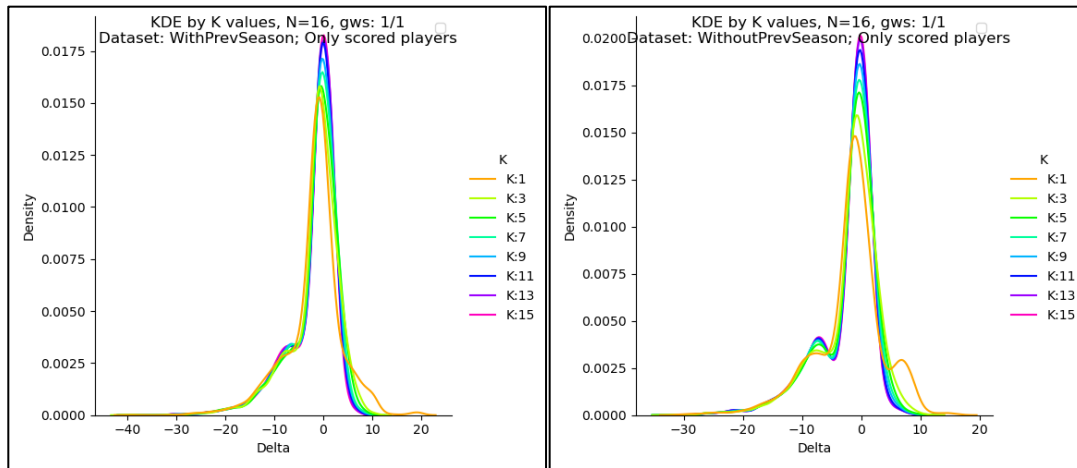
## AI Project - Fantasy Premier League Player

נציין תחילה כי הגדרנו  $\Delta := Predicted - Real$  על מנת לראות לא רק את גודל השגיאה, אלא גם את כיוונה. מהתבוננות בגרפים המצורפים מטה, נבחין כי באופן כללי ההפרשים בדיוק בין ערכי  $K$  שונים הולכים ופוחתים ככל שערכי  $N$ -גדלים, וכי בכלל המקרים המודל טעה טעויות מינוריות יחסית והתפלגות הייתה "חלקה" כפי שציפינו, להוציא פיק קטן בהערכת חסר של 5 – שככל הנראה נבע מהניקוד שמקבלים שחקני הגנה על משחק עם שער נקי. ניתן לראות כי בעת שימוש בנתוני העונה הקודמת הערכת החסר הזו (והכללית) השתפרה ככל שערכי  $N$  עלו, בעוד שללא שימוש בנתונים אלו לא היה שינוי משמעותי – בדיוק כפי שראינו קודם. נקודה מעניינת היא שבשני המקרים עבור ערכי  $N$  גדולים אין התאמה בין ערכי  $K$ -ה שהובילו לשגיאה הנמוכה ביותר עבור כלל השחקנים לעומת אלו שהניבו נקודות כלשהן – באופן עקבי ערכי  $K$ -ה המקסימליים שנבדקו הניבו את התוצאות הטובות ביותר עבור השחקנים שהניבו ניקוד. כמו כן, ראינו כי המודל נוטה להערכת חסר מאשר יתר. התנהגות דיי דומה נצפתה הן עבור שימוש ב-Dataset הכולל את ממוצעי העונה הקודמת, והן עבור זו שאינה כוללת את נתונים אלו.



התנהגות דומה הוצגה גם עבור  $N$  ימים אחרים, ולמרות זאת כאשר הרצנו את המודל הכללי על קבוצת המבחן תוך שימוש במודל זה קיבלנו תוצאות שאינן מספקות. חקרנו ומצאנו כי אומנם התפלגות השגיאה הממוצעת נראית מספקת, אך זאת תחת הסתכלות על כלל השחקנים. התפלגות תוצאות השחקנים מוטה מאוד לכיוון מספרים נמוכים, כפי שניתן לראות בגרף המצורף בפרק בסיס הנתונים. לאור זאת, ייצאנו מחדש את גרף ה-KDE עבור שחקנים שקיבלו ניקוד בלבד, ומכאן ניתן היה לראות בצורה יותר ברורה כי המודל נוטה לחיזוי נמוך באופן מונוטוני, וכי באופן כללי קיימים פספוסים רבים עבור השחקנים היותר מעניינים לענייננו – אלו שהניבו תוצאות חיוביות. תופעה מעניינת שראינו שבעת הכללת נתוני העונות הקודמות השגיאות נראות חלקות יותר, ונראה למראית עין כי היא שיפרה את ביצועי המודל. אנליזה מעמיקה יותר גילתה כי השגיאות המצטברות אכן ירדו במודל זה, אם כי בערך זניח ( $< 0.1$ ).

## AI Project - Fantasy Premier League Player

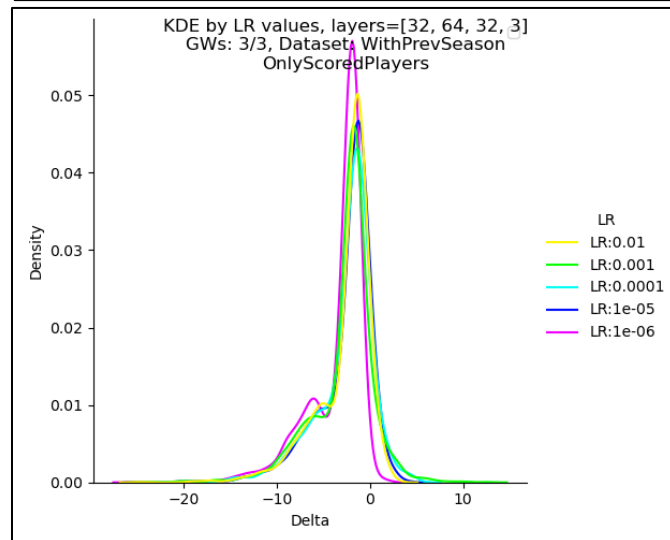
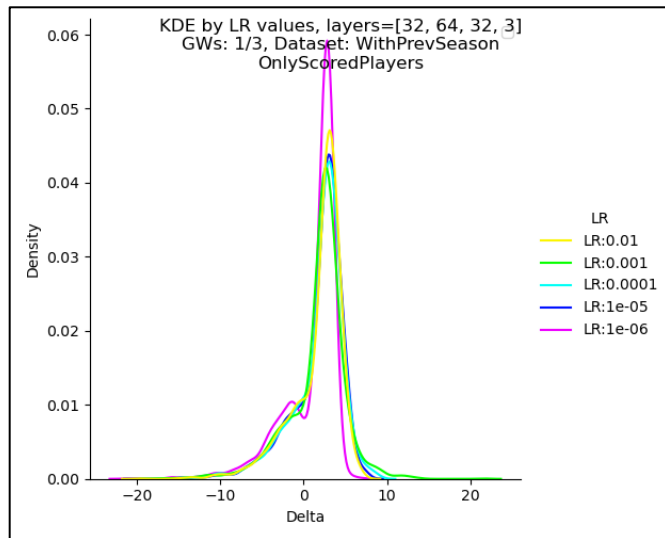
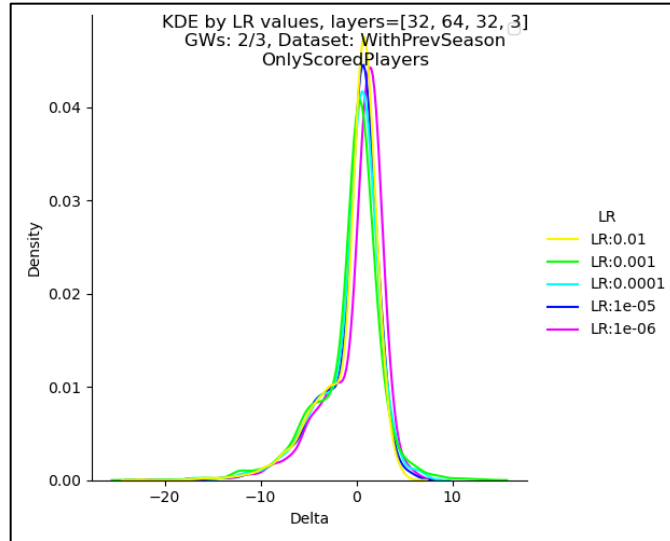


נציין כי מעבר לניסויים אלו, ניסינו לחזות את המידע באמצעות כלים שונים כמו שימוש ב-KNN בסיסי; הוצאה רנדומית של מאורעות עם ניקוד 0 מה-Dataset; שקלול תתי עצים לפי נוסחאות שונות (מרחק, ציון ממוצע של רשומות העץ); שימוש ב-CNN; שימוש ב-RNN; ושימוש ב-Multilayer Perceptron עם גדלי שכבות שונים – אך בכל המקרים קיבלנו תוצאות פחות מעניינות על קבוצת המבחן.

האחרון סיפק את הביצועים הכי טובים (למעט RandomKNNForest), לאחר שבדקנו מודלים מארכיטקטורות שונות – הן מבחינת מספר השכבות (מספר מטריצות המשקל) והן מבחינת גדלי השכבות (מימדי המטריצות). הערכים שנבדקו הם  $Layers \in [(32), (32,64), (32,64,32), (32,64,128), (32,64,128,64)]$  כאשר בסוף קיימת שכבת FC נוספת  $3 \rightarrow Layers[-1]$ , כאשר 3 הוא מספר המחזורים אותם ניסינו לבנא. ה-Optimizer שנבחר לחישוב הצעדים הוא Adam, וגדלי הצעדים נגד כיוון הגרדיאנט שנבדקו היו  $LR \in \{10^{-i} | 2 \leq i \leq 6\}$ . כל מודל אומן לאורך 300 צעדים / התכנסות ה-Optimizer (המוקדם מביניהם).

הארכיטקטורה הטובה ביותר שמצאנו הייתה תוך שימוש בממוצעי העונה הקודמת ותוך שימוש ב-3 שכבות ביניים, כאשר עבור רוב ערכי ה-Learning Rate קיבלנו תוצאות דיי דומות. הערה: עבור  $LR \in [10^{-5}, 10^{-6}]$  המודל לא הצליח להתכנס תוך 300 צעדים.

## AI Project - Fantasy Premier League Player



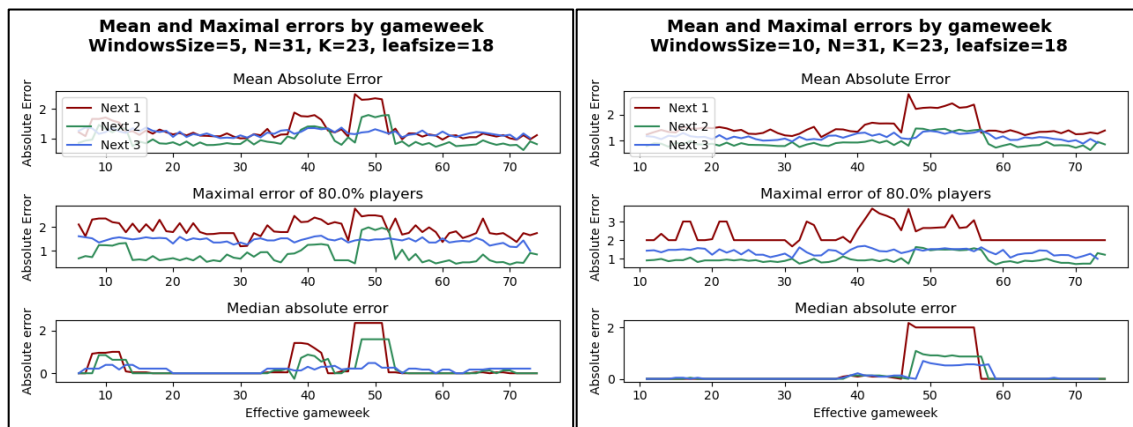
## AI Project - Fantasy Premier League Player

### 5.2.2 גישה דינאמית

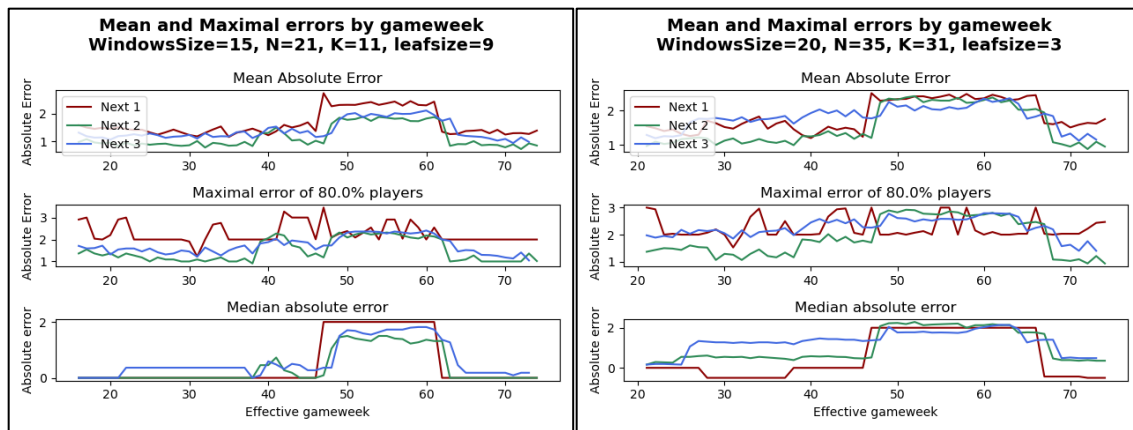
סדרת הניסויים הבאה נועדה לכייל את המודל שנבנה עבור הגישה הדינאמית, ושלבסוף נבחר, על ידי בחינת שילובים של מספר פרמטרים שונים. כפי שהוסבר בפירוט בפרקים הקודמים, המודל הנבחר כולל 3 RandomKNNForests אשר מבצעים את הפרדיקציה בהסתמכות על Sliding Window של תוצאות מחזורי העבר. נקודה משמעותית כאן היא שבסט הניסויים הזה, בחנו תחזית לשלושה מחזורים קדימה – במקום אחד בלבד. כפי שהוסבר [בתיאור אלגוריתם השחקן](#), קיימת חשיבות ל"תכנון מקדים" של הפעולות, וביצוע חוזי אמין לטווח ארוך יותר באופן טבעי מאפשר קבלת החלטות מושכלת יותר. הניסויים בוצעו על עונות 2018-19, 2019-20 באופן רציף על מנת לדמות על ההמשכיות המתוכננת עבור עונת המבחן 2020-21. בגרפים מטה מספר המחזור מתואר כ-Effective GW, שהוא שרשרת מחזורי עונות אלו בסדר כרונולוגי. הפרמטרים שנבחנו כאן הם כדלקמן:

- Window Size: כמות מחזורי העבר בהם יש להתחשב בעת ניבוי התוצאות הקרובות, כפי שהוסבר בפרקים הקודמים. הערכים שנבדקו הם  $WindowSize \in [5, 10, 15, 20]$ , כאשר וויתרנו על ערכים גדולים יותר מתוך מטרה לשמור על בידול מהגישה הסטטית.
- $N, K$ : הפרמטרים עבור כמות העצים הכללית בכל יער, וכמות העצים לפיהם יש לבצע פרדיקציה. הפרמטרים שנבחנו הם  $N, K \in [(23, 11), (31, 23), (35, 31)]$  אשר נבחרו לצורך כיסוי אזורי משמעותיים שונים שזוהו בניתוחים שהוסברו מעלה.
- Leaf size: הגודל המקסימלי של העלים בעצי הרגרסיה בעצים. הפרמטרים שנבחנו הם  $LeafSize \in [3, 9, 18]$  אשר נבחרו מתוך הנחה שככל שמנסים לחזות לטווח ארוך יותר תפיסת אנומליות תהיה משימה קשה יותר, וייתכן כי עלים גדולים יותר יניבו תוצאות משמעותיות יותר (ספויילר: ההשערה התגלתה כנכונה).

בהיבטי שגיאה אבסולוטית, ראינו כי ההבדלים המשמעותיים בין הקונפיגורציות השונות נובעים בעיקר משינוי ב- WindowSize. עבור אותו ערך WindowSize התקבלו תוצאות דיי דומות עבור שאר הפרמטרים (שוב, בהיבטי שגיאה אבסולוטית גרדיא; בהמשך נבצע ניתוח מעמיק יותר), ולכן החלטנו להציג נציג עבור כל ערך שכזה בחלוקה לפי שבועות המשחק – שכן זו יחידת המבחן שלנו כעת.



## AI Project - Fantasy Premier League Player



תופעה מעניינת אשר משתקפת בצורה מובהקת בהתבוננות על השגיאה החציונית ועל MAE, היא שבאופן עקבי קיימת קפיצה משמעותית באזור מחזור פתיחת העונה השנייה (2019-20), והשגיאה נמשכת בערך לאורך כל החלון עד אשר המודל חוזר רק על משחקים מהעונה הנוכחית.

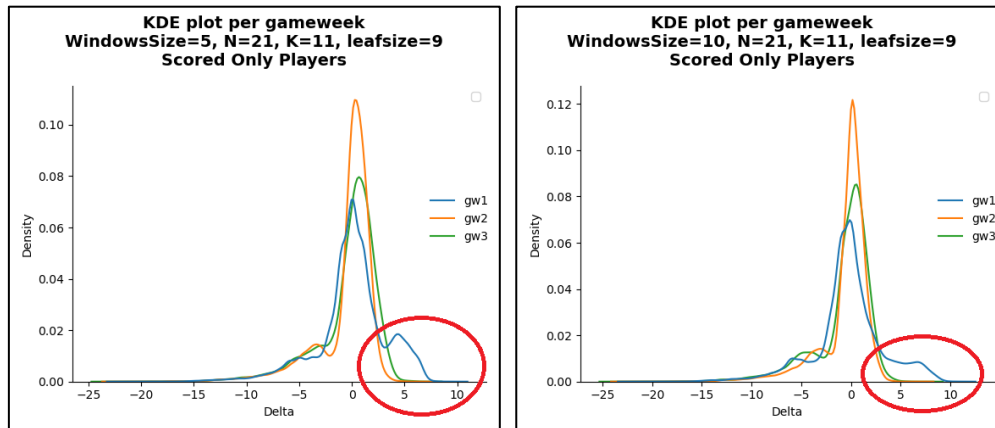
תופעה נוספת אשר עולה כאן, כמו גם בתוצאות שיוצגו בהמשך, היא שברוב המקרים השגיאה עבור מחזור אחד קדימה גדולה מאשר שניים / שלושה קדימה – דבר שעל פניו נוגד את ההיגיון. אנו מאמינים כי הסיבה לכך טמונה במבנה לוח המשחקים, ובפרט השינויים בין משחקי בית/חוץ אשר ידועים כמשפיעים על שחקנים – אך כפי שנראה בהמשך הבדל זה משתקף בעיקר בכמות הניקוד האבסולוטית, אך לא היחסית.

אנומליה מעניינת היא הקפיצה בשגיאה שמתחילה באזור מחזור 8 של עונת 2019-20, שניתן לראות בגרף של WindowSize=5. ניסינו להבין מה יכול היה לגרום לכך, וכשהסתכלנו על לוח המשחקים שמחנו לגלות שאכן היו שם אירועים יוצאי דופן – באותו מחזור, שלוש מבין הקבוצות הגדולות והחזקות בליגה הפסידו לקבוצות חלשות, ולא כבשו ביחד אפילו לא שער אחד. מנצ'סטר יונייטד הפסידה לניוקאסל, תאומתה תאוות הבצע מנצ'סטר סיטי הפסידה לוולבס, וטוטנהאם חטפה שלישייה מברייטון. זאת ועוד, אסטון וילה, קבוצת התקפה בינונית מינוס (באותה עונה), הבקיעה חמישייה. שילוב זה ככל הנראה הטעה את המודל בצורה מסויימת כל עוד היה בחלונות, והיות ובמקרה זה שמרנו רק 5 מחזורים אחורה משקל אנומליה זו היה לא מבוטל.

אם כן, החלטנו להתמקד על  $WindowSize \in [5, 10]$ , ולנתח את התפלגות השגיאות באמצעות KDE. נציג כעת השוואה עבור קונפיגורציה ספציפית לדוגמה, אך רוח הדברים הייתה דומה גם בשאר המקרים. כמסקנה מניתוחים קודמים, על מנת לראות בצורה טובה יותר את הנתונים דילגנו על שלב ההתפלגות של כלל השחקנים ונציג ישירות את התפלגות השחקנים שקיבלו ניקוד השונה מאפס.

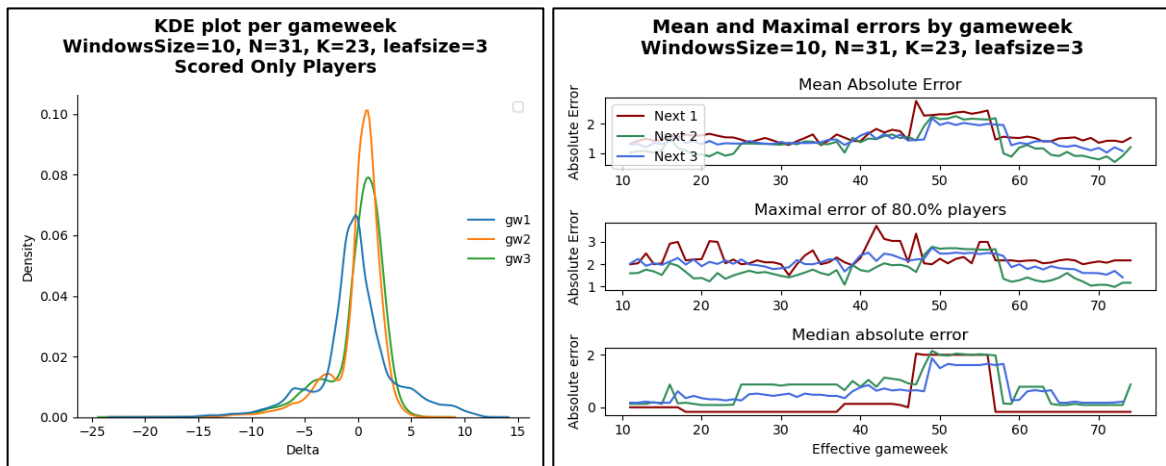


## AI Project - Fantasy Premier League Player



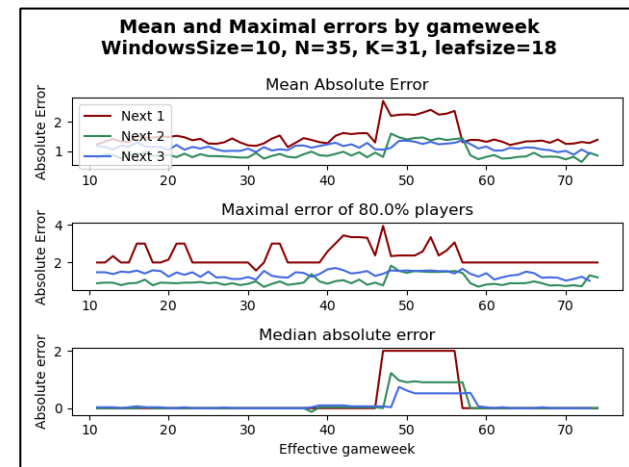
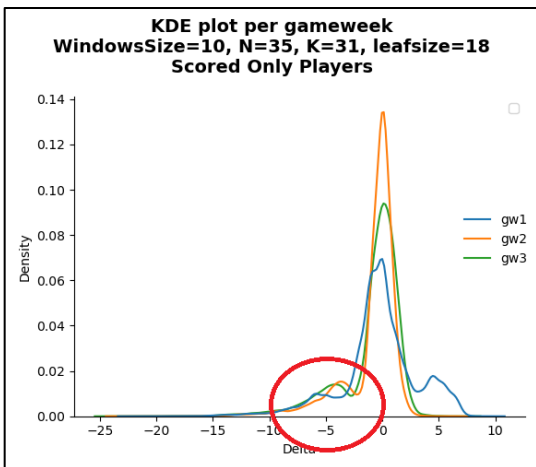
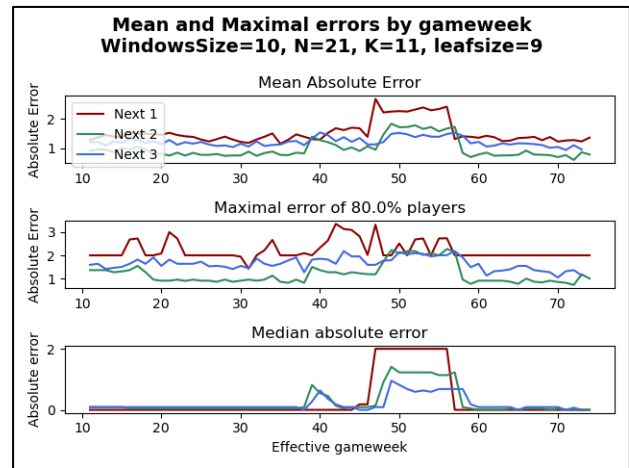
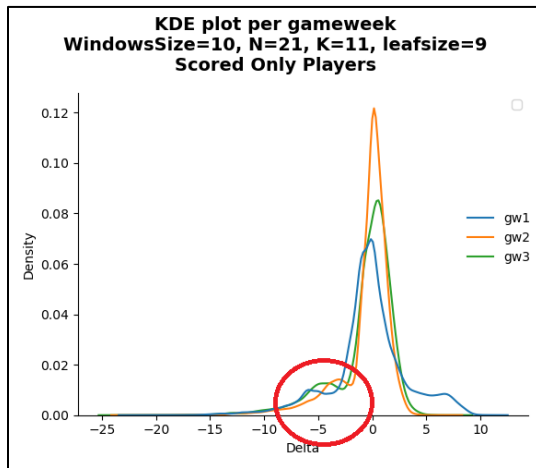
ניתן לראות הבדלים משמעותיים בהתפרסות השגיאות עבור שגיאה חיובית (מוקפות באדום) – ככל הנראה כתוצאה מ-Overfitting למידע בעת שימוש בחלונות בגודל 5. אומנם השגיאה הממוצעת עבור WindowSize=10 הייתה לרוב קצת גבוהה מאשר WindowSize=5, אך היות ובמשחק זה פיזור השגיאות משמעותי לא פחות ואולי אף יותר, החלטנו להמשיך את הבדיקות עבור WindowSize=10.

כעת עמדה בפנינו השאלה אילו פרמטרים לקבוע עבור כל תת מודל, עבור חיזוי לטווחים השונים. עבור מחזור אחד קדימה, ראינו באופן עקבי כי התוצאות הטובות יותר מתקבלות עבור Leafsize=3, כאשר הקוניפוגרציה המתאימה ביותר בהיבט זה הייתה עבור N=31, K=23:



## AI Project - Fantasy Premier League Player

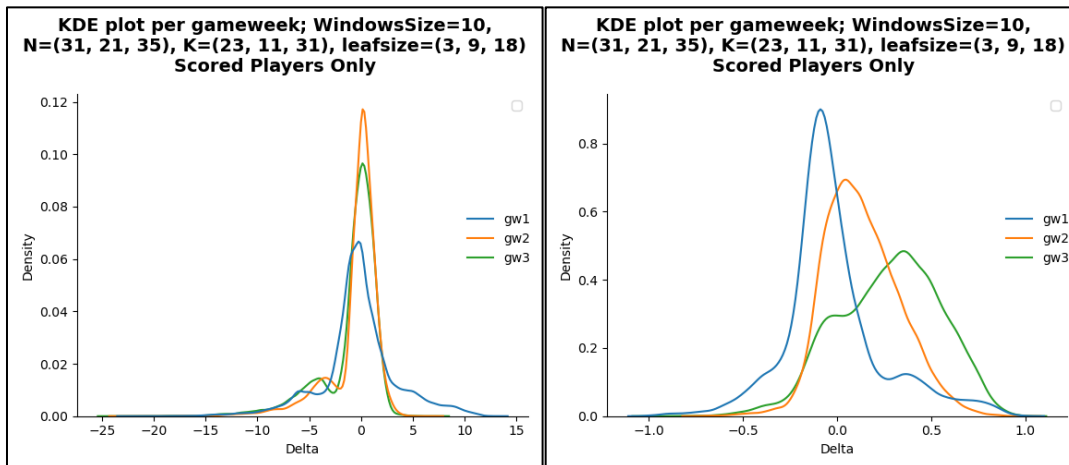
עבור שניים ושלושה מחזורים קדימה, בלטו שתי קונפיגורציות עיקריות:



ההפרשים היו קטנים ביותר, אך בסופו של דבר מצאנו כי פיזור השגיאות של הקונפיגורציה העליונה מתאים יותר עבור חיזוי של שני מחזורים קדימה, והתחתון עבור שלושה. על גרפי ה-KDE מסומנת נקודת המפנה עבור gw2 שגרמה לבחירת הקונפיגורציה העליונה עבור שני משחקים קדימה. עבור שלושה מחזורים קדימה המשימה הייתה מורכבת יותר, והיות וההבדלים בבחינת ה-KDE היו מינוריים ביותר לטובת הקונפיגורציה התחתונה, הכרענו את הכף באמצעות ה-MAE, אשר היה נמוך יותר ועקבי יותר בקונפיגורציה השנייה.

## AI Project - Fantasy Premier League Player

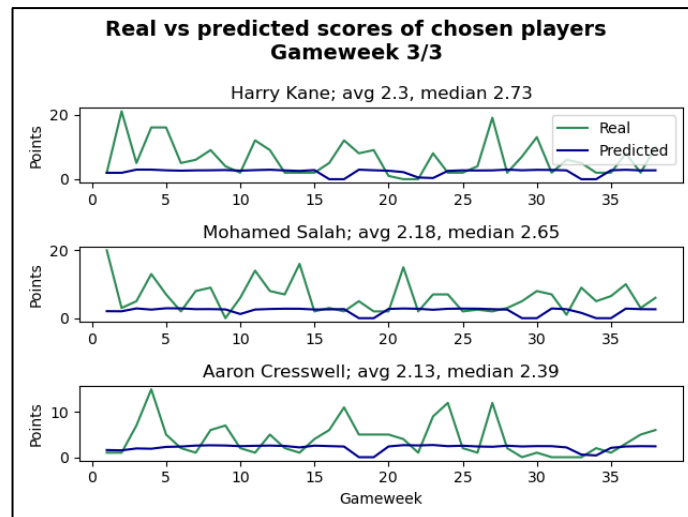
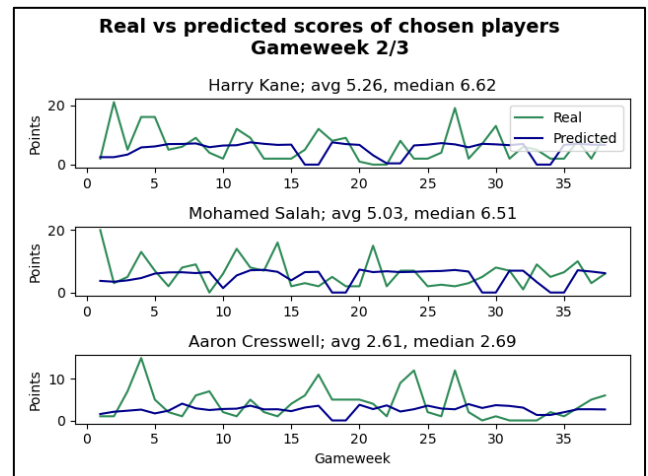
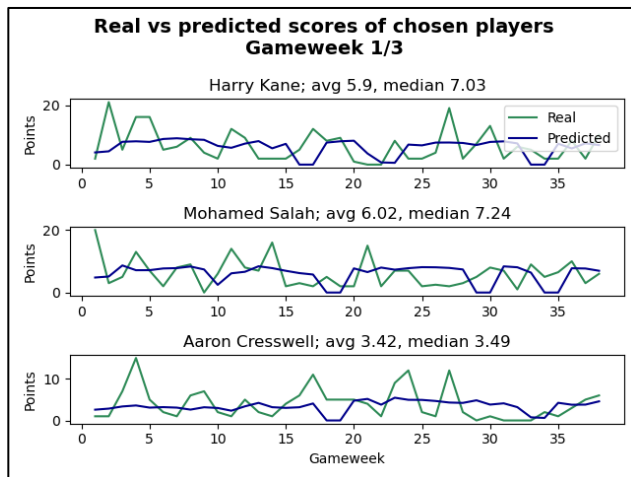
בסופו של דבר, המודל שהתקבל הניב תוצאות אשר סיפקו אותנו וכפי שנראה בהמשך, הובילו לתוצאות מכובדות של הפרויקט כולו. קל לראות כי פיזור השגיאות כעת טוב משמעותית מאשר בגישה הקודמת. נקודה חשובה שיש לשים אליה לב, היא שמאופן פעולת המשחק לעיתים מספיק לדעת מה צפויים להיות הציונים היחסיים של השחקנים, ולא דווקא את הציון המדויק שלהם (אם ברונו קיבל 4 וכל השאר 2, ההבדל באם המודל יבא לו 2 ולשאר 1 אינו קריטי ברוב המקרים, אם כי הוא כן בעל חשיבות בתכנון המשחק). לכן ועל מנת לתת משנה תוקף לפרמטרים שנבחרו, התבוננו שוב בהפרשים כאשר הפעם כל מחזור נורמל לטווח  $[0,1]$  (תחזיות לחוד וציוני אמת לחוד) והבטנו בהפרשים, אשר הניחו את דעתנו בהקשר לפערים שראינו בין החיזויים לטווחי החוזי השונים.



## AI Project - Fantasy Premier League Player

### 5.2.3 ניתוח תוצאות אמת

לאחר שאימנו את המודל על עשרת המחזורית האחרונים של עונת 2019-20, החלנו לבצע פרדיקציות על המחזורים של עונת 2020-21. התוצאות הועברו לאלגוריתם בחירת השחקנים, אשר תוצאותיו הסופיות יפורטו בהמשך. הסתקרנו עד כמה קרוב המודל למציאות וכיצד התנהג על עונה זו, ולפיכך הוצאנו את תחזיותיו על 2 מאריות הליגה ומהמובילים בניקוד בעונה זו (קין וסלאח), וכן על שחקן אפרורי, לכל אחד משלושת המחזורים. ניתן לראות שאומנם הדיוקים לא היו בשמיים, אך לרוב המגמות היו מדויקות – וכן צפי הניקוד אכן הולם את שציפינו – השחקנים החזקים היו לרוב עם תחזיות גבוהות משל ידידנו אהרון, אשר ציוניו הגבוהים היו בעיקר אנומליות במשחקים חריגים.



## AI Project - Fantasy Premier League Player

מכיוון שממוצע על כל השחקנים לא כ"כ רלוונטי – הרוב מקבלים ניקוד שנע בין 0-2 (לא משחקים, או שלא עושים הישג מיוחד על המגרש המזכה בניקוד). לכן, שווה להתמקד בשחקנים בעלי ניקוד גבוה. השוואה מעניינת נוספת היא של 30 השחקנים אשר אצלו צברו את הניקוד הגבוה ביותר (בבחינת חיזוי למחזור אחד קדימה) לעומת ה-30 שצברו זאת במציאות – הן מבחינת הניקוד עצמו, והן מבחינת מיקומם בטבלאות אלו:

Predicted rank	Name	Real rank	Predicted scores	Real scores	Predicted rank	Name	Real rank	Predicted scores	Real scores
1	Mohamed Salah	3	228.60	231	16	Andrew Robertson	13	151.39	161
2	Harry Kane	2	224.27	242	17	Wilfried Zaha	42	149.70	136
3	Bruno Fernandes	1	217.81	244	18	Kevin De Bruyne	33	147.53	141
4	Heung-Min Son	4	214.12	228	19	Neal Maupay	104	146.76	105
5	Patrick Bamford	5	193.36	194	20	Ollie Watkins	11	140.77	168
6	Jamie Vardy	6	191.76	187	21	Danny Ings	54	140.30	131
7	Sadio Mané	8	177.43	176	22	Jack Grealish	43	139.64	135
8	Marcus Rashford	9	176.67	174	23	Trent Alexander-Arnold	15	139.17	160
9	Pierre Aubameyang	55	174.72	131	24	Mason Mount	25	137.77	147
10	Roberto Firmino	32	171.22	141	25	Leandro Trossard	51	133.28	132
11	Timo Werner	59	166.38	128	26	Richarlison de Andrade	73	132.96	123
12	Raheem Sterling	20	164.63	154	27	Emiliano Martínez	7	131.11	186
13	Dominic Calvert-Lewin	12	160.41	165	28	Aaron Cresswell	21	129.82	153
14	Alexandre Lacazette	57	152.09	129	29	Jack Harrison	14	126.55	160
15	Stuart Dallas	10	151.49	171	30	Nicolas Pépé	84	122.89	114

ניתן לראות כי בשמינייה הראשונה זוהו פגיעות מרשימות ביותר – כל שמונת השחקנים הראשונים לפי המודל שלנו אכן היו בתשיעייה הראשונה בעונה זו, ומוקמו בסדר כמעט זהה! נוסף על כך, כל השחקנים שדורגו ב-15 המקומות הראשונים בעונה זו אכן הופיעו אצלו בטופ 30, כאשר 10/15 מתוכם דורגו כטופ 15 ע"י המודל שלנו. המודל זיהה בסך הכל 3 שחקנים שאינם טופ-60 כחלק מרשימה זו, כאשר שניים מהם בחמישייה שסוגרת את רשימה זו.

לאחר תחקיר מעמיק יותר על המתרחש, נראה כי במקומות מסוימים הצלחנו לתפוס "רצף טוב" של שחקנים. למשל, Dallas, שחקן אלמוני למדי בעונה ראשונה בליגה הבכירה, הפתיעה עם ניקוד מרשים מאוד והצלחנו "לתפוס" אותו בזמן! הפרדיקציה המוצלחת שלו מרשימה, ואנו סבורים שהדבר התאפשר אך ורק בזכות המודל הדינמי עם החלון הרץ – הרי בתחילתה עונה, לא היו עליו שום נתונים שמעידים שהוא הולך לספק מספרים וביצועים מרשימים, והדבר "נבנה" משחק אחר משחק.

תופעה דומה קרתה עם Martinez, במקום ה-27 אצלו. אומנם רחוק מהדירוג המקורי שלו, אך נראה כי המודל הצליח לתקן תו"כ ריצת העונה ולהבין שהוא בתקופה מעולה, ולכן תיעדף אותו.

מצד שני, את שחקני ארסנל האמכזבים – כמו Lacazette ו-Aubameyang – לא הצלחנו "לתפוס" בזמן. כלומר, דירגנו אותם די גבוה, כנראה על סמך הישגים המעולים מסוף העונה שעברה, אך התיקון הגיע מאוחר מידי וכבר צפינו להם יותר מידי נקודות. אכן מאכזב, אך מראה שבסופו של דבר המודל לא יודע – וגם לא צריך! – להתמודד עם כל אנומליה חריגה. מדוע? טוב, כי באן מודל ההיפ שלנו נכנס למשחק. לדוגמא, מודל הייפ הפיק

## AI Project - Fantasy Premier League Player





סנטימנט שלילי לגבי Aubameyang במהלך העונה, מה שהרחיק אותו מההרכב דרך אלגוריתם השחקן. מערכת משומנת!

התייחסנו לכך בפרק של פיתוח עתידי ומסקנות, אך נוכל רק לשער שאולי המודל ידע להתמודד טוב יותר עם עמדות מסוימות או רצפים חיוביים, מאשר עמדות אחרות במגרש או רצפים שליליים. ועדיין, הטבלה הזו יותר ממרשימה, והייתה יותר ממספקת כל שחקן פנטזי חובב ומקצועי.

## AI Project - Fantasy Premier League Player

### 5.3 ניסויים וכיוון מודל המדיה

ראשית, נציג דוגמא לציוצים מהעת האחרונה, שהעברנו את אותו התהליך לצורך הדגמה של סינון וסיווג:

סנטימנט	השחקן	עבר את הסינון?	ציוץ
-	-	לא. האלגוריתם זיהה שמדובר ב-2 שחקנים, ולכן סינן ציוץ זה.	 <b>Sky Sports Premier League</b> @SkySportsPL · Oct 30 A powerful double from Reece James and a Jorginho penalty helped Chelsea strengthen their grip on top spot in the Premier League with a 3-0 win away to a Newcastle side who will now be dreading a relegation battle. Match report 📰📺
חיובי	Aaron Ramsdale	כן. מופיע שחקן אחד בלבד.	 <b>Sky Sports Premier League</b> @SkySportsPL · Oct 30 Two early goals and some Aaron Ramsdale heroics helped Arsenal continue their strong form with a 2-0 win at Leicester on Saturday. Match report 📰📺
חיובי	Hwang Hee-Chan	כן. מופיע שחקן אחד בלבד (השם השני הוא של המאמן, אותו אלגו' השמות לא זיהה)	 <b>Sky Sports Premier League</b> @SkySportsPL · Oct 30 🗣️ "It's not about just the goals he scores but the work he does." Bruno Lage praises Hwang Hee-Chan's work rate for Wolves and says the striker offers much more than just scoring goals.
חיובי	Aubameyang	כן. מופיע שחקן אחד בלבד (השם השני הוא של המאמן, אותו אלגו' השמות לא זיהה)	 <b>Sky Sports Premier League</b> @SkySportsPL · Oct 29 🗣️ "That's what I want, a happy Auba." Mikel Arteta praises Pierre-Emerick Aubameyang's leadership and energy that has lifted the #AFC team.

לצורך המחשה של הדאטה שעליו אנו עובדים, נציג ענן מילים:



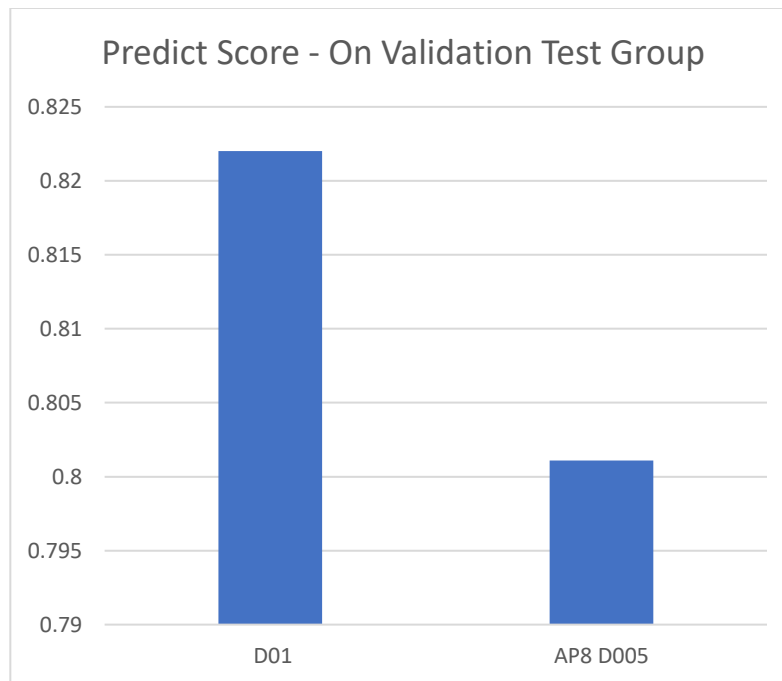
מילים בולטות: שמות הקבוצות, שמות שחקנים בולטים, מעברים, קיץ, טורנירים, עמדות, עסקה, מאמן, חתם. בעיקר שמות העצם. המילים הבאות ברשימה הם שמות תואר כמו "נהדר" ו-"יוצא מהכלל" או "מאוכזב" ו-"חלש".

## AI Project - Fantasy Premier League Player

### 5.3.1 ניסוי 1: stop words

ראשית, רצינו לבדוק השפעות העיבוד המקדים וה-embedding על המודל. לשם כך השתמשנו במודל שבחנו, כאשר השינוי היה רק האם להשמיט או להשאיר את ה-stop words.

הופתענו לגלות כי דווקא **להשאיר אותם** – בניגוד למה שלמדנו (בקורס מאיצים) – נתן תוצאות טובות יותר! שיערנו שבבעיות שהמטרה שלהם היא ניתוח סנטימנט, יש למילים אלו משמעות חשובה. אכן נתקלנו במעט מקורות שמאוששים השערה זו<sup>11</sup>



אכן התוצאה הפתיעה תחילה, כי הייתה נוגדת את מה שלמדנו, אך ממחקר קצר (ומהגיון בריא) מסתבר שכלל האצבע של "מחק Stop Words" אינו תמיד נכון, ובעיקר במשימות כמו שלנו – ניתוח סנטימנט.

<sup>11</sup> <https://medium.com/@limavallantin/why-is-removing-stop-words-not-always-a-good-idea-c8d35bd77214>

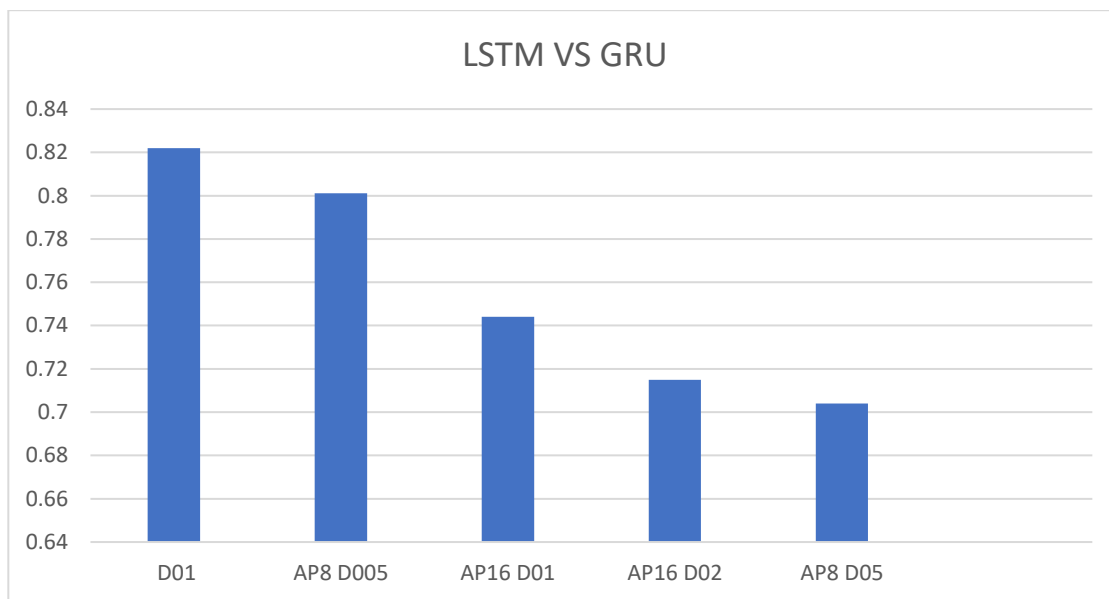


## AI Project - Fantasy Premier League Player

### 5.3.2 ניסוי 2: LSTM vs GRU

הנדבך המרכזי במודל לחיזוי סנטימנט היה קודם כל לבחור את היחידה הבסיסית. לאחר מחקר ובדיקה (בעיקר מלמידה דרך הקורס של למידה עמוקה על מאיצים חישוביים) החלטנו לבדוק מודלים מבוססי רצף וזיכרון – RNNים בכלליות. מכאן, המשכנו לחקור ולקרוא. התלבטנו והחלטנו לבחור 2 מודלים לחיזוי:

1. LSTM, אותו הכרנו – והרחבנו. זה המודל שבסוף החלטנו לממש.
  2. GRU, עליו קראנו ולמדנו במהלך הפרויקט<sup>12</sup> <sup>13</sup>. מהמאמר שקראנו, הרושם שהתקבל הוא שזהו המודל העדיף למשימה.
- בקצרה על **GRU** – עובד בשיטה דומה מאוד ל-LSTM, אך פשוט יותר (פחות מורכבות בשערים של כל יחידה וללא יחידת זיכרון). ע"פ המאמרים שבחנו (רפרנסים בתחתית העמוד), הסקנו כי:
- GRU יעיל יותר, מהיר יותר ונראה שעובד טוב עם רצפים יחסית קצרים. זוהי שיטה חדשה יותר.
  - LSTM איטי יותר אך עובד טוב יותר עם רצפים ארוכים יותר, ונראה כי מזהה קשרים "רחוקים" יותר בין מילים.
- בדקנו בניסוי את המודל שלנו עם שכבות שונות. לצורך העניין, כאשר מצוין GRU 128-64 מדובר על מודל בעל 2 שכבות GRU ובניהם שכבת Pooling.



הופתענו לגלות כי GRU הפסיד. ייתכן כי הסיבה היא שחלק ניכר מהציוצים הם יחסית ארוכים (בהם LSTM מצטיין), אך התקשנו להשים את היד על הסיבה המדויקת. השערה נוספת הם שכבות הביניים (pooling, dropout) אך גם עם כיוון של פרמטרים אחרים בהם – היחס בין התוצאות ניצח, והמודל של LSTM 128-64 ניצח.

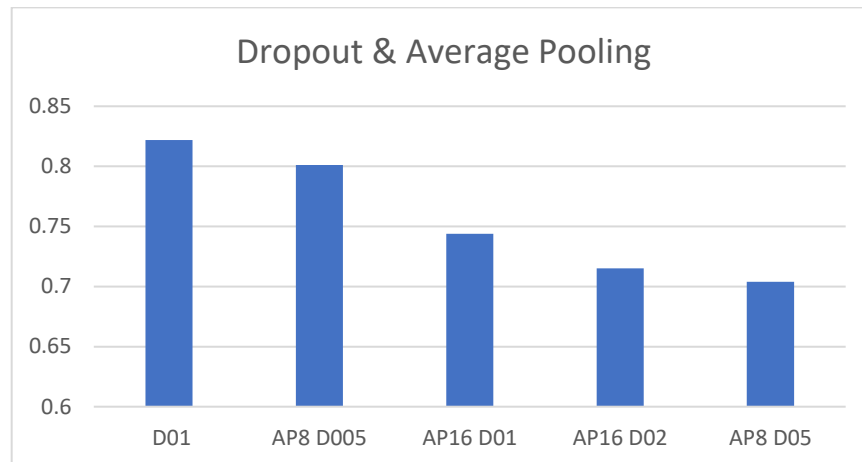
<sup>12</sup> <https://arxiv.org/abs/1702.01923> - מאמר על רשתות אלו, ורשתות נוספות

<sup>13</sup> <https://arxiv.org/pdf/1412.3555v1.pdf> - מאמר על רשת זז

## AI Project - Fantasy Premier League Player

### 5.3.3 ניסוי 3: Dropout, Pooling

החלטנו למקד את הניסוי הזה על המודל של LSTM 128-64 ועליו לעשות את מרבית הבדיקות. פה כיווננו את ההיפר-פרמטרים של שכבות ה-dropout וה-pooling, ושיחקנו עם מס' השכבות שלהם. חלק מהתוצאות היו מחרדות, והחלטנו להזניח ולא להציג אותם. המשמעות של AP-Average Pooling, Dropout, המספרים של drpoput מציינים 0.1, 0.05 וכו'.



ראינו כי dropout = 0.1 ניצח, ללא שכבות pooling.

עם אחוז דיוק של 0.82, המודל הסופי שניצח:

```
def make_model(first_layer):
    nlp_model = models.Sequential()
    nlp_model.add(first_layer)
    nlp_model.add(keras.layers.LSTM(128))
    nlp_model.add(keras.layers.LSTM(64))
    nlp_model.add(layers.Dropout(0.1))
    nlp_model.add(keras.layers.Dense(2, activation='sigmoid'))
    optimizer = keras.optimizers.Adam()

    nlp_model.compile(loss='categorical_crossentropy',
                      optimizer=optimizer,
                      metrics=['accuracy'])
    return nlp_model
```

ניתן לראות את הבניה בשכבות של הרשת הסופית שבה עבדנו. השכבה הראשונה, first layer, היא שכבת ה-embedding.

מודל זה עבר את תהליך האימון המלא, ולאחר מכן השתמשנו בו כמסווג עבור מודל ההיפ. נציין שנבחן, אך עם תוצאות רעות מאוד:

- שכבות קונבולוציה במקום LSTM. – בדקנו את Conv1D שהניב תוצאות רעות.
- שכבות Pooling נוספות (Max למשל), שהניב תוצאות רעות.
- Optimizers שונים מ-Adam שהניב תוצאות פחות בהרבה (הדבר התיישב עם סדרת ניסויים שעשינו בקורס מאיצים).

## 5.4 תוצאות, ניסויים וכיוון אלגוריתם השחקן

### אמל"ק: תוצאות המודל הסופי!

הפרמטרים של המודל הטוב ביותר הם:

משקולת מחזור 3	משקולת מחזור 2	משקולת מחזור 1	הפעלת ווילד קארד ופרי היט	הפעלת בנצ' בוסט	הפעלת טריפל קפטן	חילוף וחילוף כפול	קבוצות במרחב החיפוש	הייפ
0.1	0.4	1	30	12	13	0 – לא לשמור חילוף	15	2

- משמעות המספרים הם חישוב ניקוד הסף להפעלה.
- עבור הייפ ומשקולות המחזורים מדובר על משקולות על הנתונים בטבלת הפרדיקציה.

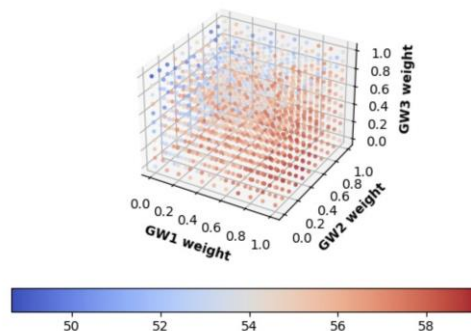
### התוצאה הטובה ביותר היא... 2382 נקודות!

- מדובר על מיקום בטופ 90 אלף של המשחק, מתוך 8.5 מיליון שחקנים!
- כלומר, אנו נמצאים ב-1.05% השחקנים הטובים ביותר!

### פרמטרים לכיוון השחקן:

- צ'יפים. כל הצ'יפים נקבעו לפי ריצות על עונת 19-20 (לא העונה של הסימולטור), ומשם בחרנו את הערך הקרוב לערך המקסימלי שמתקבל. בחרנו בעונה זו מכיוון שגם היא הסתיימה בעונת קורונה, והייתה בעלת נתונים דומים.
- הפרמטרים הדפולטיביים בקובץ params.
- הייפ. משקל על פיצ'ר ההייפ. ביצענו מספר הרצות עם משקולות שונים – כדי לראות השפעה על העונה (לא ככוון פרמטר). בדקנו מספר משקולות של הייפ, וככל שנותנים לו ערך גבוה יותר – עד 2 – התוצאות עולות במעט בכל עלייה. המשקולת נותנת משמעות גדולה יותר עבור הפרדיקציה למחזור הקרוב.
- מספר קבוצות התחלתיות (נבדק ע"י חוטים שרצים במקבל) - ככל שנגדיל את מספר הקבוצות כך אנו מתחילים עם קבוצה סבירה יותר אך ככל שיש יותר קבוצות, כך ההרצה נהיית איטית יותר.
- ביצענו הרצות עם פרמטרים גבוהים, שלקחו שעות, ומרגע שזיהינו שהפרשים די נמוכים – קיבענו את הפרמטר.
- חילוף וחילוף כפול – התוצאות הכי טובות התקבלו ע"י שימוש בחילוף כל מחזור, ולכן השתמשנו בפרמטר שקובע שעל השחקן עדיף לא לחסוך חילוף.
- כיוון המשקולות בוצעו ע"י ריצה מעונה קודמת והיה חלק מכיוון הפרמטרים – בחנו מס' משקולות שונים בהרצות שונות:

Season scores by GW weights

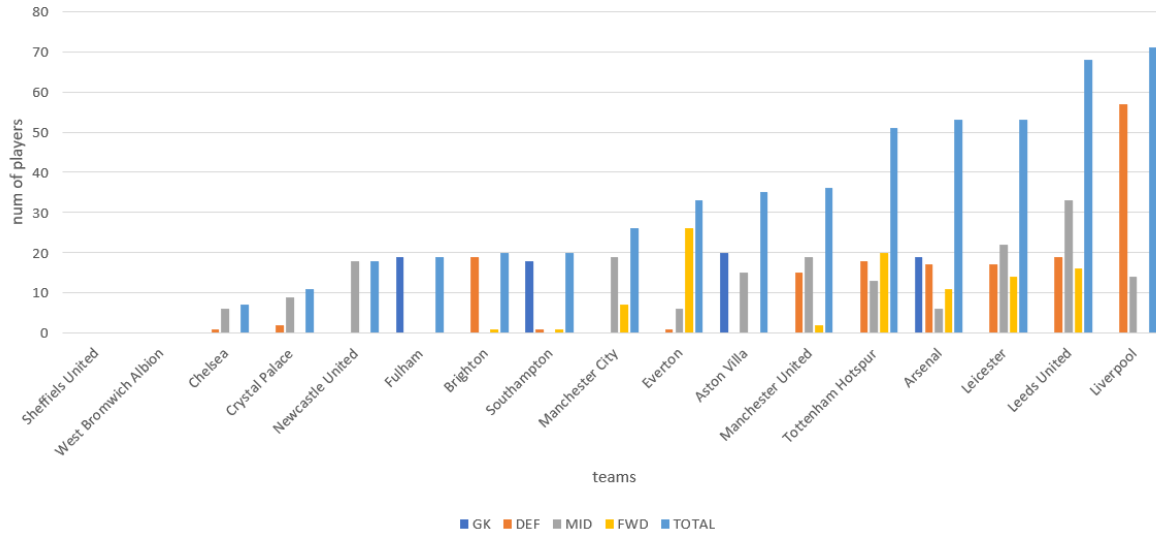


## AI Project - Fantasy Premier League Player

### נתונים וסטטיסטיקות של השחקן:

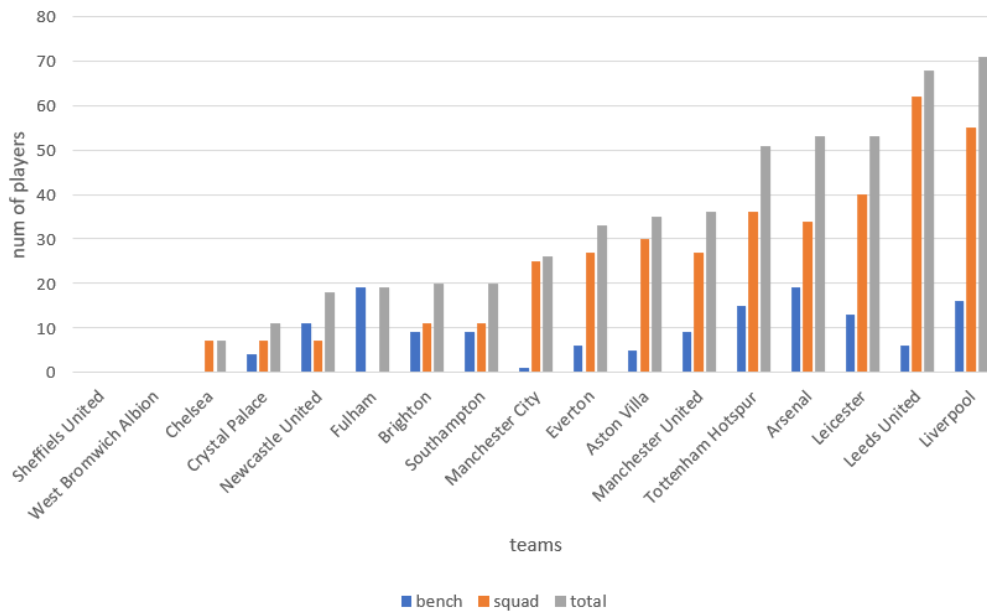
- מספר שחקנים שנבחרו מכל קבוצה:

number of players picked per team and position



- מספר שחקנים שנבחרו מכל קבוצה ספסל וסגל

number of players picked per team bench and squad

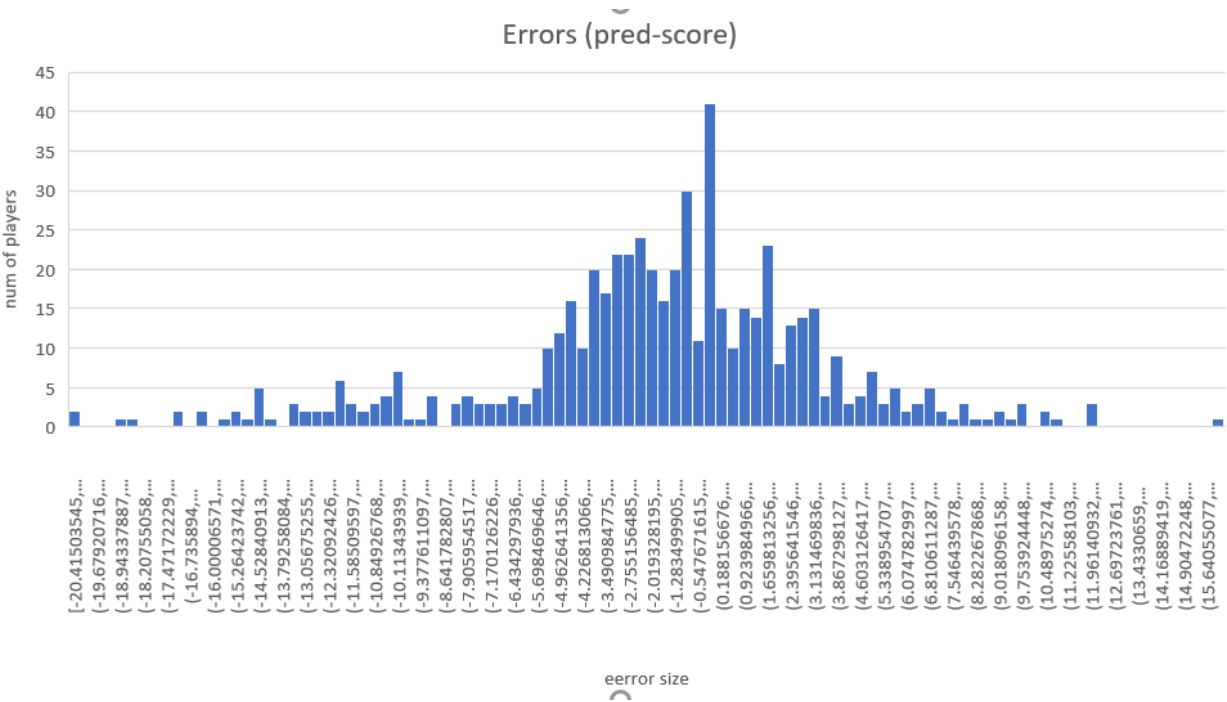


AI Project - Fantasy Premier League Player

ובצורה טבלאית:

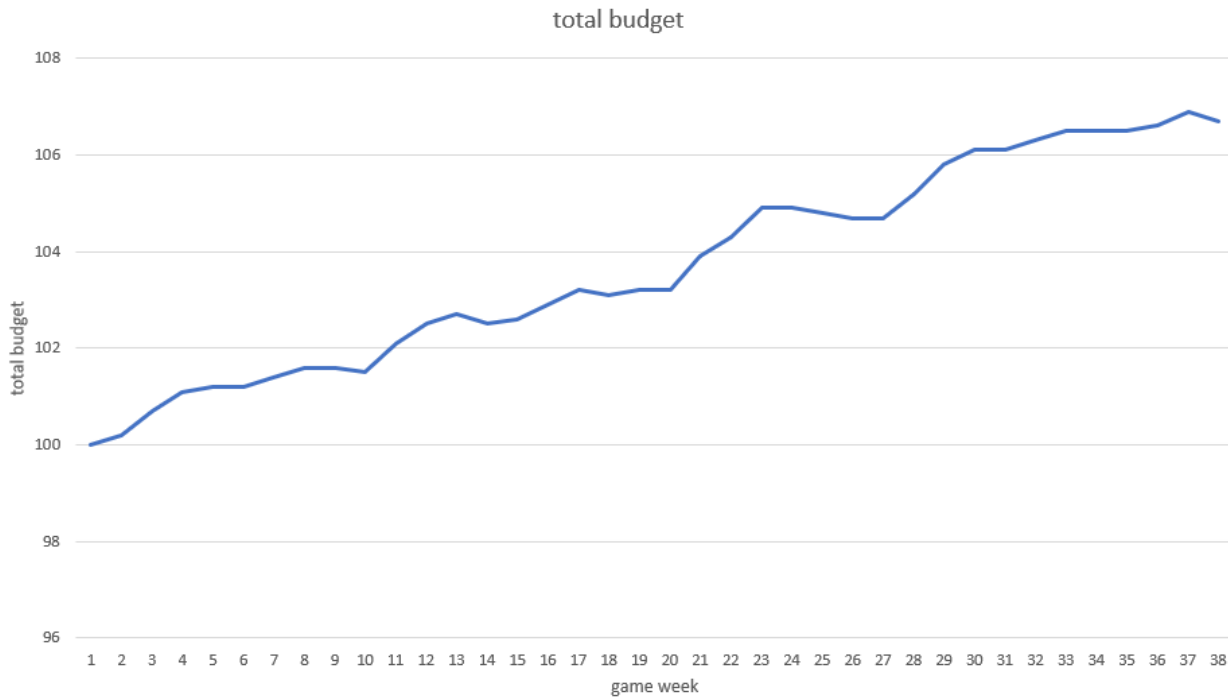
team	GK	DEF	MID	FWD	TOTAL
Sheffields United	0	0	0	0	0
West Bromwich Albion	0	0	0	0	0
Chelsea	0	1	6	0	7
Crystal Palace	0	2	9	0	11
Newcastle United	0	0	18	0	18
Fulham	19	0	0	0	19
Brighton	0	19	0	1	20
Southampton	18	1	0	1	20
Manchester City	0	0	19	7	26
Everton	0	1	6	26	33
Aston Villa	20	0	15	0	35
Manchester United	0	15	19	2	36
Tottenham Hotspur	0	18	13	20	51
Arsenal	19	17	6	11	53
Leicester	0	17	22	14	53
Leeds United	0	19	33	16	68
Liverpool	0	57	14	0	71

ערך שגיאה של מספר שחקנים:



## AI Project - Fantasy Premier League Player

- תקציב הקבוצה כתלות בזמן:



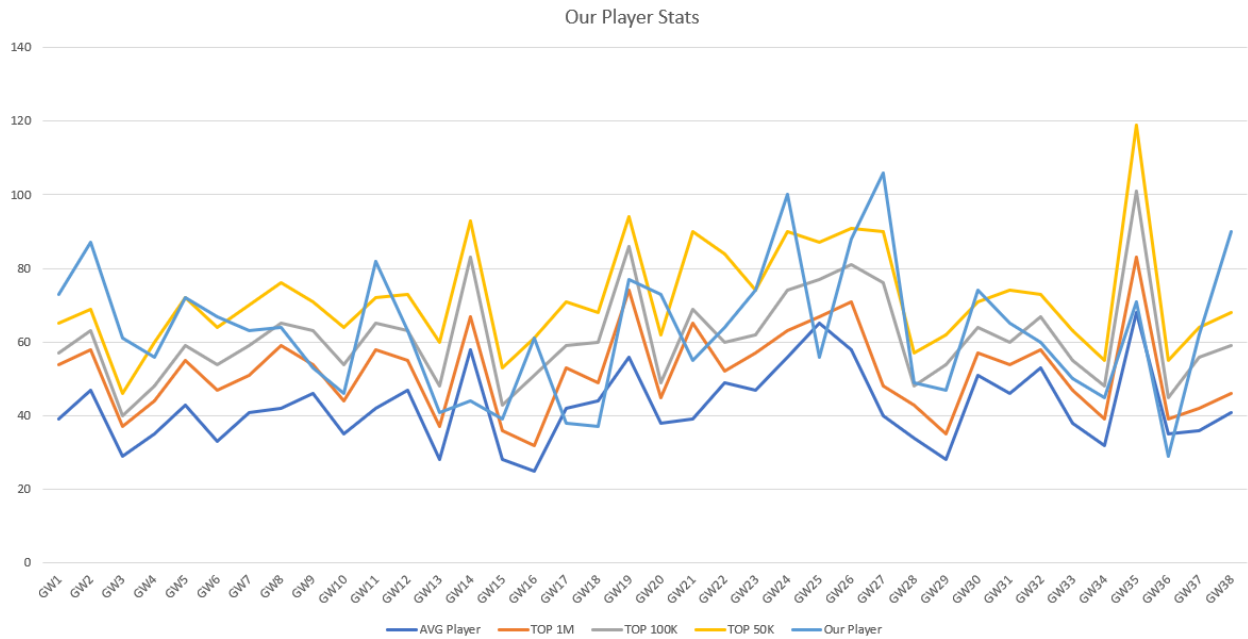
פירטנו על נושא זה בפרק הבא, ולא התמקדנו בנושא ניהול התקציב והשחקנים כמניה. אך נראה כי הבחירות המושכלות הובילו גם לעליית ערך הסגל בצורה מרשימה.

- ממוצעים בולטים:

ניקוד ממוצע לקפטן	ניקוד ממוצע לשחקן הרכב	ניקוד ממוצע לשחקן בספסל	ניקוד ממוצע לשחקן סגל
7.53	4.7	2.34	4.01

## AI Project - Fantasy Premier League Player

- ניקוד השחקן לעומת ממוצעי שחקנים בולטים:



TOP 50K	TOP 100K	TOP 1M	Our Player
2731	2325	1975	2382

ניתן לראות את ההצלחה של השחקן שלנו. מתחקיר של הנקודות לפי מחזור, זיהינו "נפילות" יחסיות רק במחזורים החריגים – כלומר מחזורים כפולים, כמו במחזור 35, שלשחקן שלנו לא היה דאטה לגביהם ובעיקר דרך פעולה. פירטנו על כך בנקודות להמשך.

### 6. מסקנות

- **אלגוריתם דינמי, המתחשב בדאטה עדכני, קריטי במשחקים וניתוח מסוג זה.** לדאטה עדכני קיים משקל משמעותי מאוד עבור חיזוי תוצאות עתידיות. התבוננות על המידע באופן סטאטי וניסיונות חיזוי על סמך שורות סטטיסטיות ללא הקשר כרונולוגי הניב תוצאות פחותות מאשר עדכנו באופן דינאמי. יחד עם זאת, כפי שראינו, יש גם חשיבות להסתכלות על מידע וותיק על מנת להפחית את משקלן של אנומליות, כפי שראינו עבור  $WindowSize = 5$ .
- **המידע שנבחר בקפידה עבור יצירת ה-Dataset התברר כנכון ומדויק למשימתנו.** הסתכלות עליו באופן עקבי תרם למודל לקשר בין מאפייני השחקנים במחזורים השונים וליצור תמונה טובה יחסית של העתיד לבוא, כפי שראינו שמודל ה-Hype חיפה על חיזויים מופרכים עבור אובמיאנג.
- **חיזוי מדויק של התוצאות התברר כמשימה בעייתית, שכן זיהוי אנומליות אינו דבר של מה בכך (בפרט כשמדובר בביצועים של בני אדם) – אך עם זאת, זיהוי הדפוסים הכלליים והמגמות התאפשרו ואף בצורה מוצלחת, כפי שהעידה טבלת ה-Top30 ותוצאות ההרצה הכלליות.**
- **מודל המדיה - ההייפ והכשירות – תרמו בצורה אדירה ליכולות המודל.** הכוח של הרשתות החברתיות והדאטה שהגיע מהטקסט הוא חשוב ביותר ונותן אינפוט נוסף למשחק: קל וחומר בעונת קורונה.
- **בחנו שיטות שונות לפתירת בעיית החיפוש באלגוריתם השחקן. מכיוון שמרחב האפשרויות הוא עצום, הגענו למסקנה – שהוכחה כנכונה – שהשיטה הטובה ביותר היא לרוץ בצורה מקבילית על קבוצות רנדומליות, שמנסות למקסם את התוצאה ע"י רצף פעולות לביצוע.**

ובאופן כללי, בעיות חיזוי מהעולם האמיתי דורשות דאטה רב. ההצלחה במשימה ובהנדסת מערכת איכותית נובעת משילוב של מודלים שונים ותקיפת הבעיה מזוויות שונות.



### 7. הצעות לשיפור, פיתוח עתידי וסיכום

#### שיפורים למודל חיזוי הנקודות:

- יכול היה להיות מעניין לראות איך מודל חיזוי התוצאות היה מתמודד עם חלוקה נוספת לפי עמדות: אילו היינו משכפלים את המודל שהוצג ומאמנים כל חלק על **עמדה אחרת** (שוערים, מגנים, קשרים וחלוצים). היות ולכל עמדה אופי שונה, ייתכן כי חלוקה נוספת ברמה זו הייתה משפרת את ביצועי המודל ומונעת השענות על מדדים פחות רלוונטיים לשחקנים מסויימים. בנוסף, ניתן להשתמש באותה תשתית ולהוסיף באופן חלק נתונים הרלוונטיים לעמדה זו בלבד, שכן הנתון על מספר הצלות אשר קריטי לשוערים לא רלוונטי כלל לחלוצים, כמו מספר שערים במקרה ההפוך.
- ניתן לנסות להוסיף לזמן החיזוי נתונים עדכניים נוספים, אשר התווספו בין שבועות המשחק. בתצורה הנוכחית, הקלט לחיזוי עבור שבוע  $i$  הוא סטטיסטיקות המשחקים כפי שהיו בשבוע  $i - 1$  בלבד. ניתן לעבות את וקטור התכונות באמצעות מידע עדכני שנלקח משחקנים אחרים בפנטזי לדוגמה.
- בהמשך לכך וכפי שהוסבר בפרק [תיאור מודל החיזוי](#), אנו סבורים כי ארכיטקטורות של למידה עמוקה, ו-RNN בפרט, יכולות לשפר את דיוק המערכת בצורה לא מבוטלת. היות ומודלים אלו תוכננו למציאת קשרים עמוקים בין התכונות באופן רציף, ייתכן כי חרף המגבלות הטכניות של מודלים אלו ניתן לבצע להם אדפטציה שתתמודד בהצלחה עם סוג המידע שהוצע בסעיף הקודם.
- מעניין לראות כיצד בניה של מודל אשר מטרתו למזער את השגיאות היחסיות עבור כל מחזור, במקום ניסיון למזער שגיאה אבסולוטית, היה משפיע על תוצאות הפרויקט.

#### שיפורים והצעות למודל ההיפ:

- לנסות להרחיב את טווח הקלט – כלומר ציוצים בעלי יותר משם אחד של שחקן, ולנסות להבין את ההקשר במשפטים יותר מורכבים.
- הדאטה מטוויטר הגיע עם נתונים על כל ציוץ, מעבר לציוץ עצמו: כמות רטוויטים, כמות לייקים ועוד. נוכל לבחון מודל שממשקל כל ציוץ ומשמעות שלו לפי כמות הלייקים.
- לדוגמא, המודל שלנו היום מקבל משפט ומוציא סנטימט. עבור המשפט "מאמן ליברפול אמר במסיבת העיתונאים סלאח שחקן נהדר, והוא הולך לשחק בכל משחק של ליברפול!" יש כמובן סנטימנט חיובי חזק, אך גם עבור המשפט "ווליהוק שחקן צעיר ונהדר, והוא יוביל את ניוקאסל להישגים" סנטימנט חיובי. הציוץ הראשון יכול לקבל אלפי לייקים, כי מדובר בקבוצה אהודה ושחקן מוערך, ואילו הציוץ השני – פחות. נוכל למשקל כל ציוץ לפי כמות הלייקים בפועל שנוצר לו ע"י הרשתות החברתיות.
- לנסות לנתח ציוצים והודעות שהגיעו לא מגופים רשמיים, אלא מ"ידועני פנטזי". זה נשמע מצחיק, אך יש עמודים של ידוענים בתחום הפנטזי, עם עשרות אלפי עוקבים. חלקם אפילו בעלי מקצוע מתאים (אנשי הסתברות או מתמטיקה) שמנתחים את הפעולות שלהם, מפרסמים המלצות ועוד. החלטנו הפעם להתחשב רק בידיעות רשמיות ומהימנות, אבל יהיה מעניין לשלב עוד תת מודל עבור ידועני פנטזי.
- וכמובן, להמשיך לכוון פרמטרים, לבדוק רשתות עמוקות שונות ועוד. אין לכך באמת סוף ותמיד אפשר לקרוא מאמרים נוספים בתחום ולהשתפר.

## AI Project - Fantasy Premier League Player

### שיפורים והצעות לאלגוריתם השחקן:

- **תקציב.** שחקנים מנוסים, מנסים לתכנן את החילופים שלהם מעבר לניקוד השחקנים, אלא גם לפי עליית או ירידת הערך שלהם. מחיר השחקנים משתנה, ואם הצלחנו "לתפוס" שחקן זול לפני שכולם רוכשים אותו, נרוויח! כלומר לשחקן תתכן עלייה במחיר. אם הוא ברשותנו, ואז נמכור אותו – נרוויח. אם נרצה לרכוש אותו לאחר העלייה, נפסיד. כך אפשר לשחק עם התקציב, ולרוב שחקנים טובים (כמו שלנו 😊) מסיימים עם תקציב גבוה מ-104-103. במשחק, לרוב, שחקנים עולים במחיר אם יש להם ביקוש ומשתמשים רבים במשחק מכניסים אותם. וגם ההפך הוא הנכון, כלפי ירידת ערך. נוכל בעתיד לתכנן מנגנון לניהול תקציב, שמנסה למשקל את ההחלטה לחילוף גם לפי שווי ערך השחקנים.
- התחשבות בשבועות בעייתיים – אם נוכל לקבל אינדקסיה לקראת פגרת נבחרת, נוכל אולי לחשוב על החילופים בצורה שונה ולהיזהר עם שחקנים שיוצאים לתקופה עמוסה (ועלולים להפצע). במציאות, הדבר שכיח מאוד, ויהיה נחמד אם נוכל לשלב זאת (דרך מודל המדיה).
- שבועות משחק מיוחדים: מחזור כפול או מחזור ריק. בגלל הקורונה, נדחו מעט משחקים עקב התפרצויות בקבוצות. הדבר הפגיע ולא היה אפשר להיות מוכנים לזה מראש. אך לאחר מכן, שובצו המשחקים החסרים במחזור עתידי, מה שיצר מחזור כפול. מחזור כפול יצר מצב ששחקנים קיבלו ניקוד על 2 משחקים – זוהי סיטואציה מושלמת להשתמש בטריפל קפטן! נציין שגם פה, נוכל לדעת את המידע דרך מודל המדיה ודיווחים מהטוויטר. אבל נזכור כי העונה ששיחקנו הייתה מעט חריגה (קורונה, בידודים ודחיות משחקים) ולא סביר שמחזורים כפולים יחזרו באופן תדיר.
- שינוי ההייפ תו"כ העונה. בתוצאות ניתן לראות כי המשמעות של הייפ במהלך העונה חזק הרבה יותר מאשר בתחילת העונה. לכן, היינו רוצים לבצע ניסוי שנותן משמעות גבוה להייפ ממחזורי אמצע העונה עד לסוף העונה, ופחות בתחילת העונה. ההשערה שלנו היא שבתחילת העונה, יש גם "הייפ מזויף" סביב שחקנים חדשים, וקשה מאוד לתפוס מי יהיה טוב ומי לא. ככל שהעונה מתקדמת, הציטוטים והאמירות על השחקנים הם בקשר ישיר להישגים שלהם (במשחקי הקבוצה, או במשחקי נבחרות ואימונים למשל) ולכן מכאן ההסבר המתקבל.

### לסיכום, הפרויקט היה מאתגר, מלמד ובעיקר – כיף!

נגענו ולמדנו עולמות רבים ושונים: אלגוריתמיקה מסקרנת וניתוח שלה, טכניקות מדעיות של ניתוח נתונים ודאטה, מודלים של למידת מכונה ולמידה עמוקה ועוד.

היה מרתק ומלמד – המרדף אחרי המודל המושלם, לבוון פרמטר נוסף, עוד ניסוי קטן כדי לשפר את הנתונים... אבל התוצאות בהחלט מספקות, ולטעמנו מרשימות מאוד.

## AI Project - Fantasy Premier League Player

### 8. נספח - מפרט טכני, מדריך למשתמש ולבודק

תיקיית app (התיקייה הראשית עבור אלגוריתם השחקן):

- app.py – יש להריץ עם h על מנת לקבל עזרה כלפי הפרמטרים. נקודת הכניסה של השחקן.
- Create\_param – כיוון פרמטרים עבור האלגוריתם. יש להריץ קובץ זה פעם אחת לפני הרצת המשחק.
- h- להסבר עבור הפרמטרים שניתן לכוון.
- בתיקיית logs תוכלו לראות את ההרצות האחרונות הטובות ביותר, וכמו כן הרצות חדשות ינותבו לשם.

תיקיית TPnet (התיקייה הראשית עבור מודל חיזוי הציונים):

- Runner.py – קוד ההרצה הראשי של מודל חיזוי הנקודות. מכין את המודל הסופי לריצה ע"י מילוי ה-Sliding Window בעשרת המחזורים האחרונים של עונת 2019-20, והתחלת החיזוי על עונת 2020-21. התוצאות נשמרות בקבצי csv פר מחזור תחת db/simulation/scores, ומאוחדות עם המידע שהתקבל ממודלי המדיה לכדי תוצר סופי שיועבר לאלגוריתם ניהול המשחק. בנוסף, סקריפט זה מייצא נתונים כלליים על התוצאות האמת שהתקבלו.
- על מנת להריץ את הקוד, יש פשוט לקרוא לו תוך שימוש ב-Interpreter הכולל את הספריות המצוינות מטה.
- RegTrees.py – מכיל את המודלים שנבחנו המבוססים על עצי החלטה עבור מודל חיזוי הנקודות, ובפרט את המודל הנבחר (FPLSlidingWindow).
- קובץ זה לא נועד להרצה.
- DNNs.py – מכיל את המודלים של למידה עמוקה שנבחנו עבור מודל חיזוי הנקודות.
- קובץ זה לא נועד להרצה.
- Static\_datasets.py – מכיל אובייקט מעטפת עבור ה-Dataset בגישה הסטטית.
- קובץ זה לא נועד להרצה.

תיקיית TweetModels (התיקייה הראשית עבור מודל המדיה):

- playerMediaHypeModel - מכיל את בניית והכנת מודל המדיה (הייפ)
- playerAviaibilltyModel – מכיל את ה-parser של מודל הכשירות.
- getTweet – נקודת הכניסה של ה-API של טוויטר.
- TweetCleaner – מנקה את הציוצים לקראת תהליך ה-Embedding

תיקיית utils (קבצי עזר כלליים):

- fplVars.py – קובץ משתנים גלובאליים כלליים הנועד לעשות סדר בקוד, דוגמת שימוש באותו Encoding לכל קבצי ה-csv.
- namesHandler.py – קובץ המכיל את המחלקה PlayerName אשר שומשה בפרויקט לצורך זיהוי השמות באופן חד-חד ערכי, תוך התמודדות עם תווים לא סטנדרטיים ושיבושים בשמות כפי שהוסבר [בתת פרק כתיב פונטי](#).

## AI Project - Fantasy Premier League Player

- בעת הרצת הקובץ ייווצר db/all\_players.csv אשר מכיל את שמות השחקנים, הייצוג הפונטי שלהם וה-id המתאים.
- Db-parser.py – סקריפט המייצא את קבצי הקלט עם נתונים פר שבוע, שיהוו את בסיס הנתונים שלנו. ממזג בין טבלאות ונתונים ומכין את הפרדיקציות.
- playersStatsScraper.py – סקריפט אשר אוסף סטטיסטיקות מעונות עבר, עבור שימוש ב-WithPrevSeason Dataset המשמש בגישה הסטטית.
- בעת הרצה, מייצר את הקובץ db/players\_general\_stats.csv המכיל את סטטיסטיקות השחקנים בעונות עבר.
- datasetCreator.py – קובץ הנועד ליצור Dataset אחד לגישה הסטטית, תוך איחוד נתוני העונות הקודמות שהתקבלו מ-playersStatsScraper.py.

### ספריות ייעודיות:

כלל הספריות להתקנה מופיעות בקובץ yml .

- ספריית Metaphone, קישור להורדה: [/https://pypi.org/project/Metaphone](https://pypi.org/project/Metaphone)
  - מממשת את אלגוריתם Double-Metaphone שהוסבר בפרקים הקודמים.
- ספריית unicodedata, קישור להורדה: [/https://pypi.org/project/Unidecode](https://pypi.org/project/Unidecode)
  - מאפשר נרמול של מחרוזות הכוללות תווים Non-ASCII למחרוזות ASCII חוקיות.
- ספריית googlesearch, קישור להורדה: [/https://pypi.org/project/googlesearch-python](https://pypi.org/project/googlesearch-python)
  - מאפשר חיפוש קל בגוגל וקבלת תוצאות; שומש עבור עדכוני נתונים עבור עונות עבר.
- ספריית bs4, קישור להורדה: [/https://pypi.org/project/bs4](https://pypi.org/project/bs4)
  - מאפשר Parsing קל לבקשות HTTP; שומש עבור ייצוא יצירת ה-Dataset.
- ספריית sklearn, קישור להורדה: [/https://pypi.org/project/scikit-learn](https://pypi.org/project/scikit-learn)
  - עבור שימוש במגוון המודלים והמטריקות שהספרייה הנהדרת הזו מציעה.
- ספריית Pickle, עבור שמירת פלטי המודלים.
- ספריית Matplotlib, עבור גרפים.
- ספריית Seaborn, עבור גרפים.
- ספריית Pandas, עבור עבודה מול מסד הנתונים וקבצי CSV.
- ספריית heapq, עבור מיון תוצאות בצורה נוחה.
- ספריית urllib, עבור יצירת ה-Dataset.
- ספריות עבור מודלים של למידה: Keras, Pytorch
- ספריית TextBlob