**מבוא:**

כדורגל הוא ענף הספורט הקבוצתי הפופולרי והנפוץ ביותר בעולם. במשחק משתתפות שתי קבוצות, שמטרת כל אחת מהן היא הכנסת כדור המשחק לתוך שערה של השנייה. כל קבוצה מונה אחד עשר שחקנים שמשחקים בעמדות שונות (מתחלקים לשוערים, שחקני הגנה, קישור והתקפה).  
כ-240 מיליון איש ברחבי העולם משחקים כדורגל באופן סדיר, בין אם באופן מקצועני ובין אם חובבני. במסגרת הענף פועלים כ-300,000 מועדונים מקצועיים, הרשומים בגופים הרשמיים המנהלים אותו.

בשנים האחרונות, חל שינוי במודל הכלכלי שבו מועדוני ספורט (בעיקר בכדורגל) מתנהלים. ניתן לשים לב שרוב הקבוצות המובילות בעולם משקיעות לא מעט כסף ומאמץ במעקב אחרי כישרונות צעירים (בעיקר בין גילאים 16-25) מכמה סיבות: ראשית, בטווח גילאים זה, שווי שוק של השחקנים יחסית נמוך בהשוואה לשווי שלהם בשלבים יותר מתקדמים כשיגיעו לשיא הקריירה שלהם.  
בנוסף, בטווח גילאים זה, השחקנים יכולים להתפתח מאוד מבחינה מקצועית, וכך להביא לתועלת רבה לקבוצות שלהם גם מבחינת תארים לקבוצה, וגם מבחינה כלכלית למועדון (מכירת חולצות, יותר אהדה לקבוצה וגם לאחר שמחיר הכישרונות עולה, הקבוצות יכולות למכור אותם ברווח רב).  
לדוגמא, ריאל מדריד, שנחשב לאחד ממועדוני הכדורגל המובילים בעולם, השקיע בשנת 2019/2020 כ 550 מיליון יורו ברכישת שחקנים, אשר ממוצע גילם היה 22.1.  
באופן דומה, ברצלונה השקיעו 390 מיליון יורו ברכישת שחקנים, אשר ממוצע גילם היה 23.2.

בדרך כלל, החיפוש אחר הכישרונות נעשה על ידי אנשי מקצוע (שנקראים Scouts) שמוכשרים על ידי המועדונים, ומטרתם לעקוב אחר ההתפתחות של השחקנים כדי לבדוק אם הם מתאימים לרמה שהמועדון מצפה לה.  
בפרויקט שלנו, החלטנו לנסות דרך אחרת למעקב אחרי הכישרונות ולקבל חיזוי מדויק יותר לגבי התפתחותם של שחקני כדורגל בגילאים צעירים, במטרה אולי לעזור למועדונים למצוא את הכישרונות הגדולים ביותר בעולם.

**תיאור הדרך שאנחנו מציעים לפתרון הבעיה:**

הבעיה אנו מנסים לפתור היא בעיית חיזוי, ולכן אנחנו נשתמש באלגוריתמי למידה וחיזוי

שלמדנו בקורס מבוא לבינה מלאכותית ואלגוריתמי למידה נוספים .

כזכור, בתהליך הלמידה קיימים מספר חלקים, המפורטים להלן:

1. **איסוף דוגמאות:** חלק זה היה הכי מאתגר במהלך העבודה על הפרויקט. החלטנו לאסוף מידע ממקורות שונים:  
   1) **Football Manager:** בסיס הנתונים העיקרי הינו משחק המחשב Football Manager (כאשר הנתונים שהשגנו היו מהגרסאות 2011, 2012, 2017 ו- 2018). כידוע, קיים קושי רב בחילוץ נתונים ממאגר של משחק מחשב, אלא שלאחר חיפושים רבים, הצלחנו למצוא עורך מיוחד (תוכנה) שבעזרתה ניתן לסנן שחקנים לפי קריטריונים שונים. הצלחנו להתקין את העורך ב- 4 גרסאות שונות המתאימות לגרסאות המשחקים, ובעזרתם היה אפשר לחלץ קובץ נפרד בפורמט pxml עבור כל שחקן שמכיל נתונים שונים השייכות לשחקן (כמו יכולות טכניות, פיזיות, מנטאליות ומידע כללי עבור השחקן).  
   עברנו על ששת הליגות המובילות בעולם (English Premier League, Spanish La Liga, Italian Serie A, Germany Bundesliga, French Ligue 1, Portuguese Liga NOS ) ולכל ליגה עברנו על כל העמדות האפשריות במשחק ( Goalkeeper, Center Back, Left Back, Right Back, Central Defensive Midfielder, Central Midfielder, Central Attacking Midfielder, Right Winger, Left Winger, Striker) ולכל אחת מהעמדות הנ"ל חלצנו את השחקנים (מכל הקבוצות השייכות לאותה ליגה). חלקנו את המידע שקיבלנו לתיקיות לפי העמדות של השחקנים.

2) **Transfermarkt:** אתר שמכיל מידע על כל השחקנים בעולם (גם שחקני עבר), מידע כגון: שווי השוק, מספר הופעות בעונה מסוימת, מספר שערים בעונה מסוימת, מספר בישולים בעונה מסוימת, מספר שערים שספגו (שוערים) בעונה מסוימת ומספר המשחקים שבהם שמרו על רשת נקייה (שוערים) בעונה מסוימת.  
על מנת לחלץ מידע מהאתר, כתבנו סקריפט (Scrapper) שמוציא מידע לפי קוד ה- HTML, בעזרת ספריית BeautifulSoup. באותו אופן, עברנו על ששת הליגות המובילות בעולם, ולכל ליגה עברנו על כל הקבוצות השייכות לליגה, ולכל קבוצה עברנו על כל השחקנים בקבוצה וחלצנו את הנתונים של השנים 2011-2020 (שערים, בישולים, מספר מופעים וכו...). את הנתונים שמרנו בקובץ csv כאשר כל שחקן מופיע בשורה נפרדת עם הנתונים שלו.

**עיבוד המידע ומניפולציות:** רוב העבודה שביצענו בשלב הזה הייתה הקישור בין

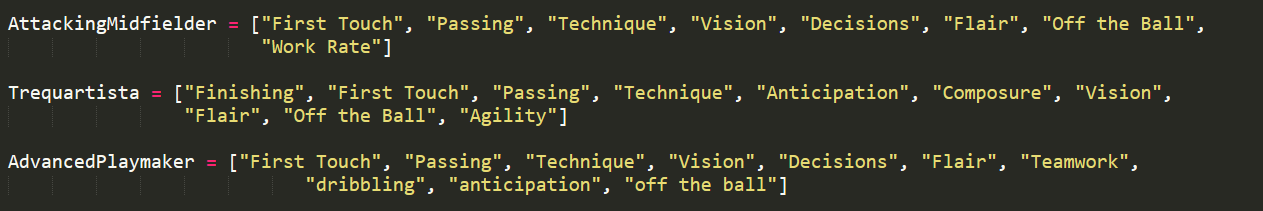
גרסאות שונות של בסיסי הנתונים של המשחקים שהשגנו והחיבור שלהם למאגרים השונים שחילצנו. ראשית כפי שהזכרנו קודם, המידע שחילצנו מהעורכים של Football Manager היה בקבצים נפרדים לכל שחקן בפורמט pxml. ולכן היינו צריכים למצוא דרך לחבר בין הקבצים השונים של שחקנים בעלי אותה עמדה וליצור קובץ יחיד עבורם בפורמט csv כדי שיהיה נוח לעבוד איתו בהמשך. כתבנו סקריפט שמטרתו לחבר בין כל הקבצים תחת תיקיה מסוימת לקובץ pxml יחיד, ואז ממיר אותו ל- csv על ידי חילוץ תכונות רלוונטיות לכל שחקן. אחרי שלב זה קיבלנו 10 קבצים בפורמט csv כאשר כל קובץ משויך לעמדה מסוימת ומכיל שחקנים שמשחקים בעמדה זו. את התהליך הבא מבצעים לארבעת הגרסאות השונות של FM ומקבלים סה"כ 40 קובץ.

האתגר הבא שנעמד בפנינו היה לחבר בין מידע ששייך ל- Football Manager לבין מידע שחילצנו מ- Transfermarkt. הבעיה הייתה מורכבת מאוד בגלל שלכל מאגר היה מזהה שונה עבור כל שחקן (למשל השם של שחקן מסוים עלול להיות שונה בצורת כתיבתו בין שני המאגרים). כדי לפתור בעיה זו, כתבנו סקריפט פייתון נוסף שעובר על כל שחקן מכל אחד מקבצי המידע של FM, ואז עובר על מאגר כל השחקנים של TM ומחשב את המרחק בין השם של השחקן שבודקים (מ- FM) לבין כל שחקן במאגר TM. על מנת לחשב מרחק בין שמות של שני שחקנים, השתמשנו בפונקציה get\_close\_matches מספריית difflib, שמתבססת על Levenshtein distance, ומקבלת פרמטר cutoff שמגדיר את סף הקרבה המינימאלי בין שמות השחקנים. בצענו מספר הרצות עם ערכים שונים של פרמטר cutoff, ובדקנו את הדיוק של התוצאות אחרי כל ריצה. ולבסוף החלטנו להשתמש בערך 0.85 של cutoff שהביא לתוצאות הכי מדויקות מבחינת התאמות בין השמות. הרצנו את הסקריפט, ואחרי שקיבלנו התאמות בין השמות של השחקנים מכל קובץ FM לשחקנים בקובץ של הTM, הוספנו לכל קובץ FM את הנתונים המתאימים לכל שחקן מקובץ הTM. את התהליך הנ"ל בצענו לכל גרסה של FM.

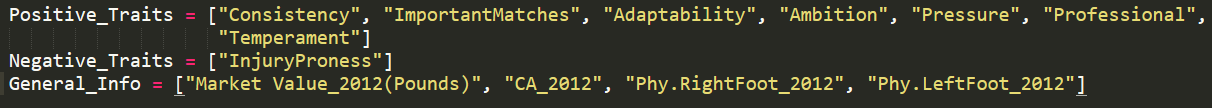
בשלב זה קיבלנו 10 קבצי csv לכל אחת מהגרסאות של FM (סה"כ 40 קובץ) אשר מכילים את הנתונים של FM וגם של TM לכל שחקן בשורה אחת. לאחר מכן היינו צריכים לחבר בין הגרסאות השונות של FM, כלומר לקבל באותה שורה מידע מ- FM11, TM11, FM12, TM12, FM17, TM17, FM18, TM18. משימה זו הייתה יותר קלה מהמשימה הקודמת כי ניתן להתאים בין שתי גרסאות שונות של אותו שחקן על פי מזהה ID ייחודי שנמצא במאגר של FM. לכן עברנו על כל הקבצים, ובכל קובץ עברנו על כל השחקנים וביצענו התאמה זו עד שלבסוף קיבלנו 10 קבצים, שכל אחד מתייחס לעמדה מסוימת ומכיל מידע מכל המאגרים שהזכרנו קודם. כמובן שהיה מספר קטן של שחקנים שלא היו זמינים עבורם כל הנתונים מכמה סיבות, למשל ב 2012 שחקו באחת מששת הליגות המובילות ואז ב 2018 עברו לליגה אחרת, או הפסיקו לשחק כדורגל מסיבה כלשהי.

**בחירת תכונות:**

לאחר שלב החיבור של הנתונים מהמאגרים השונים, קיבלנו 10 קבצי csv שכל אחד מהם מתייחס לעמדה מסוימת( Goalkeeper, Center Back, Left Back, Right Back, Central Defensive Midfielder, Central Midfielder, Central Attacking Midfielder, Right Winger, Left Winger, Striker). במשחק FM, כל עמדה (position) מתחלקת לתפקידים(roles) שונים (למשל: עמדת Central Attacking Midfielder מתחלקת לתפקידים AdvancedPlaymaker, Trequartista, AttackingMidfielder) וכל תפקיד מאופיין על ידי כמה תכונות$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ שהן הכי חשובות לתפקיד. למשל עבור תפקידי העמדה Central Attacking Midfielder, התכונות הכי חשובות הן:



בנוסף לתכונות הייחודיות לתפקידים מסוימים, קיימות תכונות כלליות, חלקן חיוביות וחלקן שליליות שחשובות לכל השחקנים ללא תלות בעמדתו/תפקידו:



רוב התכונות השייכות למשחק FM מיוצגות על ידי מספר מ- 1 עד 20, כאשר 20 הוא הציון הטוב ביותר. החלטנו לנרמל את שאר התכונות לפי סקלה זו, כך למשל עבור שווי השוק (Market Value) שערכיו נעים בין 0-100,000,000, ביצענו את הנרמול לפי הקונבנציה הבאה:

קבענו לכל עמדה חסם עליון וחסם תחתון לפי שווי השוק המקסימאלי והמינימאלי בשנות 2012-2018 (לפי אתר TransferMarkt), ואז בהסתמך על מידע זה ועל התפלגות הערכים של שווי השוק בשנים אלו נרמלנו את הערכים של שווי השוק של השחקנים למספר בין 10-20 (לא בחרנו 1-20 כי הנחנו שהערכים של שאר התכונות המתאימות לעמדה שמשחק בה שחקן מסוים גם יהיו בין 10-20 ובכך יהיה לכל התכונות השפעה זהה).

**קלסיפיקציה/רגרסיה?** לפני סיווג הדוגמאות, היה עלינו להחליט אם להגדיר את הבעיה כבעיית קלסיפיקציה בינארית (שחקן מוצלח / לא מוצלח) או בעיית רגרסיה (לתת לכל שחקן מדד הצלחה בין 0-20). החלטנו בשלב התחלתי להגדיר את הבעיה כבעיית קלסיפיקציה, כלומר שחקן יכול להיות או מוצלח או לא מוצלח.

החלטה זו נבעה מכך שגם תחת סיווג בינארי הבעיה מורכבת מאוד. סיבה נוספת לבחירה זו הייתה הקושי בלהעריך "עד כמה השחקן מוצלח?"... השאלה מאוד סובייקטיבית וסביר להניח שאין הסכמה על הגדרת שחקן כשחקן מוצלח וגם הפרמטרים של ההצלחה נבדלים בין עמדה לעמדה אחרת ולכן קשה מאוד להעריך עד כמה שחקן מוצלח. מסיבות אלו החלטתי להתייחס לבעיה כבעיית קלסיפיקציה בינארית.

**סיווג הדוגמאות:**

השוני בין העמדות מבחינת הגדרת מושג "שחקן מוצלח" עבור כל עמדה גרם לנו להשתמש במרכיבים שונים ומשקלים שונים עבור כל עמדה, בנוסף בגלל שיש שחקנים שעבורם לא קיימים כמה תכונות (למשל שחקן שסיים את הקריירה שלו בכדורגל לפני 2018 מסיבה כלשהי לא יהיו לא תכונות ב FM18) היה צורך בטיפול נפרד עבור המשקלים של המרכיבים של בעיית הסיווג, להלן פירוט של המרכיבים עבור כל עמדה:

* **Goalkeeper**:

MarketValue: 20% GoalsConceded: 20% CleanSheets: 15% CA(PA): 15% AttributesAverage: 30%

* **Right-Back**:

MarketValue: 22.5% GoalsScored: 5% GoalsAssisted: 25% CA(PA): 15% AttributesAverage: 27.5% WeakFoot: 5%

* **Centre-Back**:

MarketValue: 35% GoalsScored: 10% GoalsAssisted: 5% CA(PA): 15% AttributesAverage: 35% WeakFoot: 0%

* **Left-Back**:

MarketValue: 22.5% GoalsScored: 5% GoalsAssisted: 25% CA(PA): 15% AttributesAverage: 27.5% WeakFoot: 5%

* **Central-Defensive-Midfielder**:

MarketValue: 30% GoalsScored: 5% GoalsAssisted: 20% CA(PA): 15% AttributesAverage: 30% WeakFoot: 0%

* **Central-Midfielder**:

MarketValue: 22.5% GoalsScored: 10% GoalsAssisted: 20% CA(PA): 15% AttributesAverage: 27.5% WeakFoot: 5%

* **Central-Attacking-Midfielder**:

MarketValue: 20% GoalsScored: 15% GoalsAssisted: 22.5% CA(PA): 15% AttributesAverage: 20% WeakFoot: 7.5%

* **Right-Winger**:

MarketValue: 20% GoalsScored: 20% GoalsAssisted: 20% CA(PA): 15% AttributesAverage: 20% WeakFoot: 5%

* **Left-Winger**:

MarketValue: 20% GoalsScored: 20% GoalsAssisted: 20% CA(PA): 15% AttributesAverage: 20% WeakFoot: 5%

* **Striker**:

MarketValue: 22.5% GoalsScored: 25% GoalsAssisted: 5% CA(PA): 15% AttributesAverage: 22.5% WeakFoot: 10%

פירוט המרכיבים:

* MarketValue: The market value of the player at the end of year 2018, normalized so it gives a score between 0-20 according to the position.
* GoalsScored: Number of goals scored between 2012-2018 divided by the number of minutes played and normalized so it gives a score between 0-20 according to the position.
* GoalsAssisted: Number of assists provided between 2012-2018 divided by the number of minutes played and normalized so it gives a score between 0-20 according to the position.
* CA(PA): Current ability field from FM 2018, it’s replaced by Potential ability from FM 2012 if the player finished his career before 2018. The value of this field is divided by 10 to normalize it to a scale from 0-20.
* AttributesAverage: It indicates the average values of the attributes that are most important for the best role of the player.
* WeakFoot: A number between 0-20 that indicates how strong the weak foot of the player is.

עבור שחקנים שחסר להם אחד מהמרכיבים או יותר, נוסיף למשקל של כל אחד מהמרכיבים הקיימים איזשהו ערך מנורמל כך שהיחס החדש יהיה זהה ליחס המקורי ביניהם.

כעת, אחרי חישוב המרכיבים עבור כל שחקן, אנחנו מכפילים כל ערך של מרכיב במשקל שלו ואז סוכמים את הערכים ואז מקבלים מספר בין 0-20 אשר מייצג את אחוז הצלחתו של כל שחקן. כדי שנוכל לקבוע מי מהשחקנים הוא שחקן מוצלח, צריך לקבוע סף מספרי (לכל עמדה) כך שכל שחקן שמקבל מעל הסף יקבל סיווג 1 (מוצלח) ואחרת 0 (לא מוצלח), להלן רשימת הערכים של הסף לכל עמדה:

Goalkeeper: 13.4, Central-Back:13.6, Right-Back: 13.5, Left-Back:13.7, Central-Defensive-Midfielder: 13.4, Central-Midfielder: 13.8, Central-Attacking-Midfielder: 14.4, Right-Winger: 14, Left-Winger: 14.3,   
Strikers: 14.9.

**שלב למידה:**

השתמשנו במגוון רחב של אלגוריתמי למידה אשר יפורטו בהמשך, המטרה הייתה לבנות מסווג כך שבהינתן שחקן חדש לא מסווג, המסווג יוכל לחזות ולהניב סיווג עבור שחקן זה והאם הוא יהיה מוצלח בעתיד.  
כדי לאמן את המסווגים, השתמשנו בשיטת הcross-validation על מאגר השחקנים שלנו, בשיטה זו מאגר השחקנים חולק ל5 קבוצות, וב5 איטרציות נבחרו כל פעם 4 מתוך 5 הקבוצות לאימון המסווג, כאשר הקבוצה הנותרת שימשה כקבוצת מבחן למסווג שנבנה. בכל איטרציה חישבנו ציון להצלחת המסווג על קבוצת המבחן (לפי כמות טעויות של המסווג על קבוצת המבחן), והציון עבור הצלחת האלגוריתם נקבע על פי הממוצע של חמשת התוצאות שקבלנו מכל איטרציה.  
שיטה זו מניבה הערכה על טיב המסווג שיבנה על ידי האלגוריתם, ומה יהיו הביצועים שלו על דוגמאות שהוא לא ראה.  
הניסויים שביצענו בשלב זה כללו גם השוואת הצלחת כל מסווג כתלות בפרמטרים משתנים כתלות באלגוריתם ובפרמטרים שהו מקבל.  
על מנת לקבל הערכה שנובעת רק מאופן פעולת האלגוריתמים, ולבטל את השוני בתוצאות שיגרם כתוצאה מחלוקה שונה של הדוגמאות, הcross-validation מבוצעת על אותה חלוקה של המאגר לfolds (כלומר פרמטר shuffle כבוי).

תיאור כללי של דרך פתרון  
תיאור מערכת