**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 בתפקידים שונים (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 התפקידים האפשריים במשחק ( Goalkeeper, Center Back, Left Back, Right back, Central defensive midfielder, Central midfielder, Central attacking midfielder, Right winger, Left winger, Striker) ולכל אחד מהתפקידים הנ"ל חלצנו את השחקנים (מכל הקבוצות השייכות לאותה ליגה). חלקנו את המידע שקיבלנו לתיקיות לפי התפקידים 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 אותו תפקיד 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 לתפקיד מסוים ומכיל שחקנים שמשחקים בתפקיד זה. 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 לתפקיד מסוים 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 תכונותhidden \ \ roles \ ...

Nations name to number ..

Market value convert to number … (some information had to be cleared or changed in some other way so we can use it easily in python)

**קלסיפיקציה/רגרסיה?** לפני סיווג הדוגמאות, היה עלינו להחליט אם לסווג את הבעיה כבעית קלספיקציה בינארית או בעית רגרסיה. החלטתי שבשלב התחלתי עדיף להתמקד בסיווג בינארי, כלומר דוגמא יכולה להיות או פוגענית או לא פוגענית, ולא לנסות להחליט עד כמה היא פוגענית. החלטה זו נבעה מכך שגם תחת סיווג בינארי הבעיה מורכבת מאוד. סיבה נוספת לבחירה זו הייתה הקושי בלקבל תיוג עיקבי של דוגמאות במידה והייתי מבקשת מהמתייגים גם לתת "ציון פוגענות" לכל תגובה, ומכך שהשאלה "עד כמה התגובה פוגענית?" היא שאלה מאוד סובייקטיבית וסביר להניח שהציונים שהייתי מקבלת ממתייגים שונים היו שונים מאוד. מסיבות אלו החלטתי להתייחס לבעיה כבעית קלסיפיקציה בינארית.