✅ שלב 1 – Transfer Vectors מטרה: לבנות לכל קבוצה וקטור שמעיד אילו שחקנים היא נוהגת לרכוש.

מה צריך לעשות בפועל:

* 1.1 לטעון את קובץ transfers.csv
* 1.2 לפלטר את ההעברות הרלוונטיות (למשל exclude loans)
* 1.3 לשייך לכל מועדון את השחקנים שהוא רכש
* 1.4 עבור כל מועדון, לחשב וקטור תכונות ממוצע של שחקני הרכש (features)

איפה לגעת בקוד:

* קובץ חדש: transfers\_processing.py
* משתנה חדש: team\_transfer\_vectors: dict[str → np.array]

אלגוריתמים:

* ממוצע פשוט על תכונות → וקטור בגודל n\_features לכל מועדון.

✅ שלב 2 – Team Vectors מטרה: לבנות פרופיל של הקבוצה לפי השחקנים הנוכחיים בסגל.

מה צריך לעשות:

* 2.1 מתוך scored\_df, לקחת את השחקנים בקבוצה לפי team\_dict
* 2.2 עבור כל קבוצה ועמדה – לחשב את ממוצע התכונות של השחקנים הנוכחיים

איפה לגעת:

* קובץ processing.py או main.py
* יצירת מילון team\_vectors\_by\_position: dict[str → dict[position → vector]]

אלגוריתמים:

* ממוצע feature vector לפי קבוצה ועמדה.

✅ שלב 3 – Matrix Factorization + Cosine Similarity מטרה: למצוא קבוצות דומות לפי סגל + העברות.

מה צריך לעשות:

* 3.1 ליצור מטריצה: כל שורה מייצגת קבוצה, והעמודות הן תכונות ממוצעות של השחקנים (או רכישות)
* 3.2 לבצע Normalization (StandardScaler)
* 3.3 לחשב Cosine Similarity בין כל זוג קבוצות
* 3.4 לשמור רק את ה-k השכנות הקרובות ביותר

איפה לגעת:

* קובץ חדש: team\_similarity.py
* פונקציה חדשה: get\_similar\_teams(team\_name: str) → List[str]

אלגוריתמים:

* sklearn.metrics.pairwise.cosine\_similarity
* אופציונלי: PCA או TruncatedSVD אם רוצים להפחית מימדיות.

✅ שלב 4 – Hybrid Recommendation Model מטרה: לשלב CF (על סמך קבוצות דומות) עם CB (על סמך סגל קיים).

מה צריך לעשות:

* 4.1 לקבל את רשימת הקבוצות הכי דומות לקבוצה שלנו
* 4.2 לאסוף את שחקני הרכש של הקבוצות הדומות, ולחפש שחקנים דומים להם
* 4.3 בנוסף, לחשב similarity של שחקנים לסגל של הקבוצה לפי CB
* 4.4 לשקלל בין שני הציונים: final\_score = α \* cb\_score + (1 - α) \* cf\_score

איפה לגעת:

* קובץ score.py – הוספת cf\_score
* פונקציה חדשה: recommend\_by\_transfer\_similarity

✅ שלב 5 – הסברים ("Why this player?") מטרה: לתת למשתמש הסבר טקסטואלי פשוט מדוע שחקן הומלץ.

מה לעשות:

* לבדוק האם שחקן נבחר עקב התאמה לסגל (cb\_score גבוה)
* לבדוק האם דומה לשחקנים שקבוצות דומות רכשו (cf\_score גבוה)
* ליצור טקסט בהתאם: if cb\_score > 0.7 and cf\_score < 0.4: reason = "Fits your current squad's playing style." elif cf\_score > 0.7 and cb\_score < 0.4: reason = "Similar to players bought by similar teams." else: reason = "Recommended based on both squad fit and team similarity."

איפה להוסיף:

* קובץ score.py או recommendations.py – פונקציה: generate\_explanation(player\_row)
* בפרונטאנד: עמודה חדשה בטבלה Why this player?

✅ שלב 6 – Frontend Adjustments מה לשנות ב-UI:

* תוספת של עמודה "Why this player?" בכל טבלת תוצאות
* אולי tooltip שיסביר "Content-Based score measures how well the player fits your current squad"
* אפשרות לשליטה על α (משקל CB מול CF)