Scoutify
Football Player Recommendation System

Github

1. Sooraj Sathish *IMT2020004*
2. Kritin Potluru *IMT2020027*
3. Chinmay Parekh *IMT2020069*
4. Keshav Goyal *IMT2020101*

***Abstract*—The ever-growing popularity of football has led to an increasing demand for efficient methods of player recruitment and team building. In this paper, we propose a novel recommendation system that leverages collaborative filtering techniques to provide personalized recommendations of football players to clubs. Our system aims to assist clubs in identifying potential players who align with their specific requirements and strategic objectives, thereby enhancing their chances of success on the field.**

To build the recommendation system, we collected a comprehensive dataset comprising player attributes, performance statistics, and historical transfer information from various reliable sources. We employed collaborative filtering and machine learning algorithms to extract meaningful patterns and capture player similarities based on their past performances and career trajectories. By leveraging both user-based and item-based collaborative filtering techniques, our system intelligently matches the preferences of clubs with the profiles of players to generate accurate and tailored recommendations.

***Index Terms*—Recommendation system, collaborative filtering, football player recruitment, personalized recommendations, player profiling, user-based filtering, item-based filtering, club and player philosophy**

I. Motivation

Hindsight is a fickle thing. It's all too easy to look back on events and think that we should've seen them coming. Surely, we think, we should have known that Bitcoin was going to be a thing, that Trump would win in 2016, that the iPod touch meant the end of phones with buttons on them.

The same is true of football transfers. It's hard not to look back on moves like Andy Carroll to Liverpool, Fernando Torres to Chelsea, Alexis Sanchez to United, and Andriy Shevchenko to Chelsea and think it must've been obvious that they were doomed from the start.

Lets take an example:

In the summers of 2016 and 2017 respectively, two 25-year old wingers moved to top Champions League sides for club-record fees. Both had been performing well at their old clubs; both were valued in the £20–30 million bracket; both had overall ratings in the low-80s on FIFA. One player had scored once every two games the previous season, the other once every three. They'd even both previously had a poor spell at a top-four Premier League club, before moving abroad to secure more game time.

So, what do you think? Should it have been obvious what the respective fortunes of these players was going to be? Do you think your answer would be different if you knew their names?

One of these two was Germany international André Schürrle. He moved to Borussia Dortmund in 2016 for a fee of around €30 million. Unfortunately for him, things did not work out at the Westfalenstadion. Schürrle played fifteen times in his first season under Thomas Tuchel, scoring just twice. His market value plummeted over the next twelve months, and three years after his big-money move he was out on loan at Spartak Moscow. He retired at the end of that season.

II. Requirements

One thing we can do is to base our assessments on more objective data. If we can design and validate systems to learn patterns that are reliably associated with subsequent high performance, and prove that they work in the real world, we can start to make better decisions and take human bias out of the equation. Our football recommendation system keeps is created keeping the following constraints in mind:

- Club Philosophy
- Player Philosophy
- Budget of clubs
- Compatibility of a player with team

III. Dataset

	ID	Name	Age	Nationality	Overall	Potential	Club	Value	Weak Foot	Skill Moves	...	SlidingTackle	GKDiving	GKHandling
0	176580	L. Suárez	29	Uruguay	92	92	FC Barcelona	€83M	4.0	4.0	...	38.0	27.0	25.0
1	178518	R. Nainggolan	28	Belgium	86	86	Roma	€37.5M	3.0	3.0	...	88.0	11.0	11.0
2	181872	A. Vidal	29	Chile	87	87	FC Bayern München	€41.5M	4.0	3.0	...	84.0	4.0	2.0
3	197445	D. Alaba	24	Austria	86	89	FC Bayern München	€41.5M	4.0	3.0	...	83.0	5.0	7.0
4	195864	P. Pogba	23	France	88	94	Manchester United	€71.5M	4.0	5.0	...	73.0	5.0	6.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
104347	240558	18 L. Clayton	17	England	53	70	Cheltenham Town	€100K	2.0	1.0	...	12.0	55.0	54.0
104348	262846	♦. Dobre	20	Romania	53	63	FC Academica Clinceni	€180K	2.0	1.0	...	12.0	57.0	52.0
104349	241317	21 Xue Qinghao	19	China PR	47	60	Shanghai Shenhua FC	€100K	2.0	1.0	...	9.0	49.0	48.0
104350	259646	A. Shaikh	18	India	47	67	ATK Mohun Bagan FC	€110K	3.0	1.0	...	13.0	49.0	41.0
104351	178453	07 A. Censori	17	Italy	28	38	Arezzo	€0	2.0	1.0	...	NaN	7.0	1.0

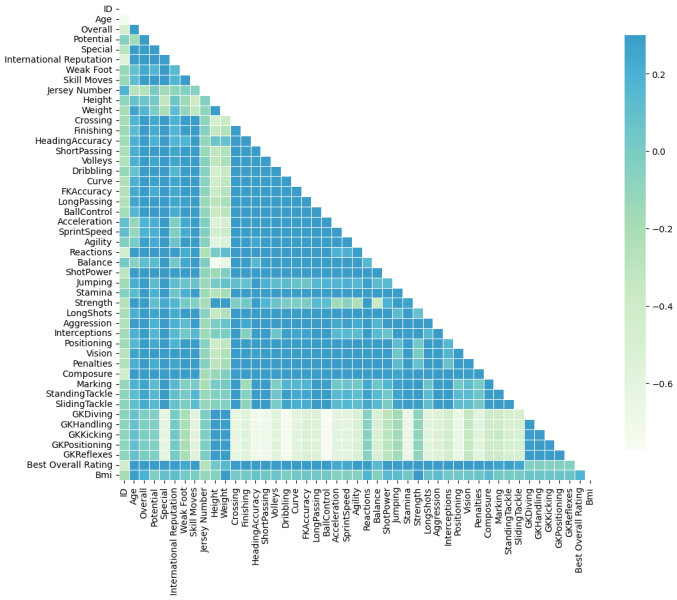
The recommendation system uses data consisting of every player from FIFA 17,18,19,20,21 and 22. This dataset consists of 104 columns:

- Player positions, with the role in the club and in the national team

- Player attributes with statistics as Attacking, Skills, Defense, Mentality, GK Skills, etc.
- Player personal data like Nationality, Club, DateOfBirth, Wage, Salary, etc.

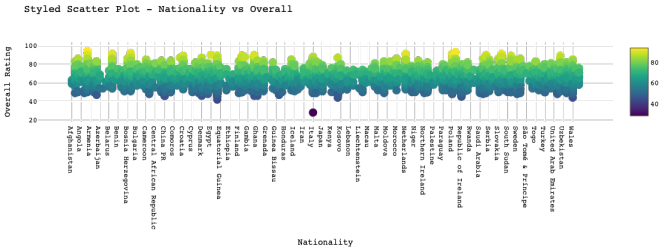
IV. Exploratory Data Analysis

A. Validity of Data

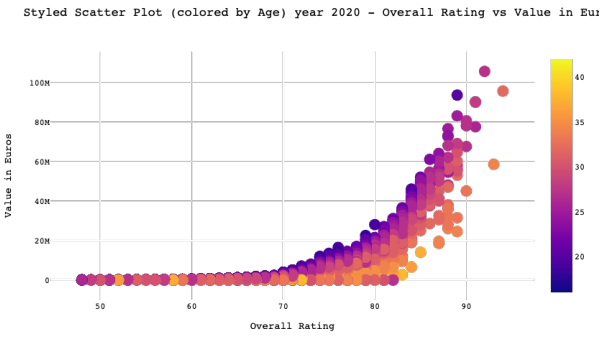


- From this correlation chart, we can see that Goalkeeper's attributes have a strong negative correlation with the attributes possessed by an outfield player.
- Players with more attacking role have higher dribbling, passing and shooting skills.
- Players having high vision have high passing and dribbling skills.

B. Interesting Observations



This scatter plot provides an overview of spread of player's rated across y-axis per nationality (x-axis). Also, this scatter plot highlights that specific countries with more players playing the game at league level hinting towards possible scouting destination. (Top recommended players in our models are from Europe, South America.)



This scatter plot illustrates the value (Millions) of a player based on their overall rating. Our focus is on top right corner. Notice that players with highest overall rating are valued more based on age grp 25-30. Players above 30 yrs are valued less despite high overall. (This pattern is realized in our models)

V. Dataset Pre-processing

- Removed unnecessary columns like Player_photo, Jersey_No., International Reputation etc
- Combined datasets from 2017 to 2022 by removing inconsistencies and stacking them vertically (ensuring same columns)
- One-hot encoding categorical variables (Strong Foot - [Left, Right])
- Cleaned Height, Weight, Valuation columns which were in string formats like "23Kg"

VI. Roadmap Models/Algorithms Used

- K-Nearest Neighbours
- BERT (NLP and Transformers)
- Linear Regression and Neural Networks
- Club Philosophy and Clustering (Final Model, Runs with Front-end)
- Modified Neural Networks and XGBoost Regression

VII. Approach 1:k-Nearest Neighbours

We started off with a basic model hoping to solve the problem of finding the closest replacement of player in a club. Players like Paul Pogba, get injured quite a lot but he is the kind of player who usually makes a big impact when he plays. Juventus would really like such k-closest players to Paul Pogba.

```
recommend_similar(176580) # Players similar to Luis Suarez
✓ 0.7s

Name:  R. Lewandowski
Overall: 91
Market Value: €€86M
Age: 30
BMI: 23.87

Name:  K. Benzema
Overall: 88
Market Value: €€53M
Age: 31
BMI: 23.62

Name:  Hulk
Overall: 80
Market Value: €€11.5M
Age: 32
BMI: 26.08

Name:  E. Cavani
Overall: 86
Market Value: €€35.5M
Age: 32
BMI: 22.43
```

A. Data

Removed all textual data and knn will be performed on all columns containing numeric data.

B. Working

- The input data is represented as a set of points in a multi-dimensional space, where each point represents an instance of the dataset.
- When a player is given as an input, the algorithm searches for the k nearest neighbours to this new instance within the dataset.
- The distance metric used is Euclidian Distance.
- The brute force method would check the Euclidian distance with all the points and so to optimize it we made use of KD Tree.

C. Analysis

Pros: The model is simple and easy to interpret.
Cons:

- Very slow as the number of data points increases because the model needs to store all data points.
- Not memory efficient
- Sensitive to outliers. Outliers also have a vote!
- The model recommends players which generally have similar ratings but it fails to take into account the team philosophy and the compatibility of a particular player with the club and team manager's style of play.

VIII. Approach 2:Transformers and NLP Models

"The value of a data point is determined by the data points around it". This was the first model where a "club philosophy" was introduced. It encompasses the playing style and character traits of a football club and its players as a whole using **textual** data.

A. Feature Engineering

"player_tags" was a text field in our dataset which described an individual's playing style and skill.

long_name	age	club	player_tags
Lionel Andrés Messi Cuccittini	32	FC Barcelona	#Dribbler, #Distance Shooter, #Crosser, #FK Specialist, #Acrobat, #Clinical Finisher, #Complete Forw...
Cristiano Ronaldo dos Santos Aveiro	34	Juventus	#Speedster, #Dribbler, #Distance Shooter, #Acrobat, #Clinical Finisher, #Complete Forward

BERT was used to create embeddings from these player tags and create a new column called "embeddings".
All rows were grouped by 'club' and inserted into a new data frame with 'club' and 'embeddings' as features. The 'embeddings' column for each grouped club was created by finding the **mean value of the embeddings** of the players present in the club.

This mean of player embeddings constitutes our first version of "club philosophy".

B. Finding Similarity Scores and Ranking Players

'similarity_score' is calculated based on cosine similarity between the player embeddings and the group club embeddings.
So now, the input is the name of the club which is looking for new players. We find the similarity score between all players in the dataset to the input club and return the list of players sorted in descending order of their similarity_score.

short_name	similarity_score
Pozo	0.83840865
M. Pjaca	0.83840865
V. Kadlec	0.83840865
C. Musonda	0.83840865
A. Najar	0.83840865
S. Özkan	0.83840865
J. Cavallaro	0.83840865
J. Dayton	0.83840865
Gustavo Lobatelro	0.83840865
D. Mitchell	0.83840865
G. Mackay-Steven	0.83840865
J. Amoah	0.83840865
R. Kishna	0.83840865

Hence, now we have a list of players who are "similar" to the input club according to the player tags in the dataset. Here, the input club was chosen as FC Barcelona. A huge number of players have the same similarity scores, something that needs to be improved in the coming models.

C. Analysis

In this model, we emphasise on the players play style and not only their overall rating. It doesn't make sense to recommend highly rated players to clubs if their play-styles do not match with the club's play-style. This model takes the first step towards fixing this problem.
Highly specific recommendations can be made to the clubs if the "player_tags" were appropriately created i.e if the "player_tags" were more specific to different players' play-styles.
This model does have its drawbacks.
A huge number of players ended up with the same similarity score due to a shortage of tags(only 83 unique tags were present).
The model didn't take into account that players might be highly similar but not good (according to their rating).

IX. Approach 3: Basic ML models

The main thought process followed here was that the improvement of a player after transferring to a club directly affects the performance of a club too. Thus clubs will want players who will improve after playing with their current players.

A. Feature Engineering

We had to create a new column , improvement(target). This was calculated by taking the difference in overall rating in consecutive years. Here we consider that every year a player transfers to a new club even if in reality he stays in the same club. This is done by assuming that club A from 2020 is different from club A in 2021. Further , some new features like average club rating , average potential of a club had to be engineered. This is because the performance of a player is dependent on how well his teammates play .

B. Models and metrics

We first picked the 35 most correlated attributes that affected the target variable, using selectKbest. We then trained a neural network and a linear regression model to predict the improvement. The metric used was MSE.

C. Prediction

Prediction of players improvement is done by taking in player attributes and only changing the club attributes of the player(average overall rating, average overall potential) to that of the club which is looking for players.

Consider player $P1$ currently in club $C1_{2020}$(club C1 in year 2020) . Now suppose club $C2_{2021}$ (club C2 in year 2021) wants him in their club . We will predict his improvement upon joining club C2 by taking a player vector and changing the club attributes of the vector from that of $C1_{2020}$ to that of $C2_{2020}$. This is because we believe that the club philosophy will not change very much in a year. That is, $P1$ with $C2_{2020}$ attributes is representative of how $P1$ plays with C2 in 2021.

Given a club we check all the players' improvements and sort them based on this improvement. We then select the 30 top players.

D. Analysis

Linear regression : 0.81

Neural Network :

Optimizer	MSE
Adam	0.8
Adagrad	0.77
Adadelta	0.78

Epochs:100, Adagrad optimizer, batch size = 128

Learning rate	MSE
0.06	0.77
0.01	0.78
0.1	0.78

E. Results / Observations

We notice that that the newly engineered features like overall rating do not affect the improvement attribute very much. Due to this all the clubs get very similar results/ recommendations.

Following is a small part of the recommendation given to FC Barcelona for year 2022:

```
'N. Larsen',
'O. Al Somah',
'B. Johnson',
'J. Mojica',
'L. Philipp',
'R. Skov',
'R. Walter',
'J. Krasso',
'C. Coady',
```

The same recommendations were given for clubs Manchester United and Manchester City.

X. Approach 4: Clustering

Having seen the results from very basic Similarity based, to the complex Bert model, we realised that the similarity based approach had more scope on the given dataset and could be extended to work with Neural Networks too.
A common drawback of previous approaches was similar recommendations to all clubs. The major factor contributing to this was lack of a sophisticated mathematical definition of Club, Player and Field Position. Our approach provides these definitions and tries to answer the question of: **HOW** a player plays instead of **HOW GOOD** a player plays.

A. Feature Engineering

What is a Position? We can define the Position (Striker, Winger, Defender etc) through a vector of attributes. It is wasteful to take a vector of all 60 attributes in the dataset, so we choose the top-k (k = 10) attributes which affect the Overall Rating of the players in that position. These top-k attributes were chosen according to their Pearson Correlation Coefficient score with the Overall rating. We call these the essential attributes of the position. We found k = 10 to be optimal as generally for all positions the Correlation score of top-10 attributes was an order higher than other attributes. We can see in the Figure that the essential attributes for Centre-Back(Defender) position have a correlation score in magnitude of 10^4 while others are in order 10^3.

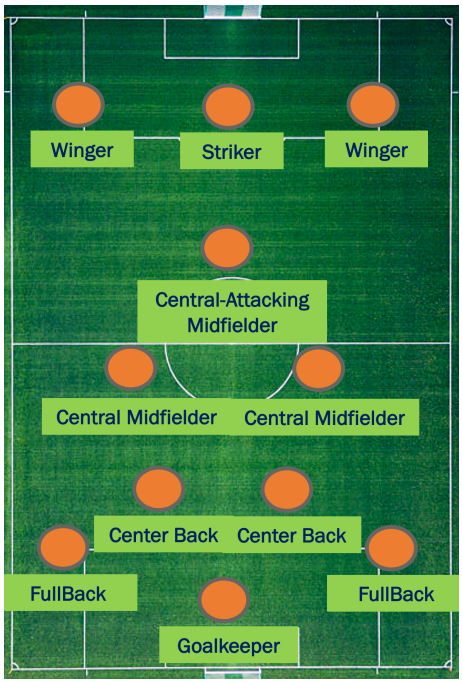
```
np.sort(fs.scores_)
✓ 0.0s
array([5.64069079e+00, 6.01175039e+01, 1.15560838e+02, 2.47649992e+02,
       3.93058587e+02, 4.06505762e+02, 9.81697813e+02, 1.11763132e+03,
       1.14061688e+03, 1.18550313e+03, 1.23847753e+03, 1.43281252e+03,
       1.45863958e+03, 2.17451588e+03, 2.75182023e+03, 2.87025967e+03,
       4.27010137e+03, 4.41289126e+03, 5.73828589e+03, 5.77270709e+03,
       6.38939382e+03, 6.72582876e+03, 6.82025961e+03, 7.05054983e+03,
       1.01421237e+04, 1.12446741e+04, 1.53094768e+04, 2.02789365e+04,
       2.03344285e+04, 2.53797729e+04, 2.61312201e+04, 3.10141541e+04,
       4.00066369e+04, 4.20135489e+04])
```

For example, the essential(top-10) attributes for Defenders were: 'HeadingAccuracy', 'ShortPassing', 'LongPassing', 'BallControl', 'Reactions', 'Aggression', 'Interceptions', 'Composure', 'StandingTackle', 'SlidingTackle'.

What is a Player? We simply define a player by the position he plays in. So if a player is a defender, we represent him with a vector of his stats in the essential attributes of the defender position. For example a defender would be represented as:

$$\text{Player}_{Def} = \begin{bmatrix} HeadingAccuracy(Player) \\ ShortPassing(Player) \\ LongPassing(Player) \\ BallControl(Player) \\ Reactions(Player) \\ Aggression(Player) \\ Interceptions(Player) \\ Composure(Player) \\ StandingTackle(Player) \\ SlidingTackle(Player) \end{bmatrix}$$

What is a Club? We identified 7 different player positions in a Club as shown in the Figure.



So to represent a single position of a Club, we took the Weighted mean of the positional essential attributes of the players playing in that position in the club. For example, let the Club C have 5 Defenders P^1, P^2, P^3, P^4, P^5, then the Defensive position for C would be the weighted mean of the player vectors of the 5 players. (These vectors would be the defensive essential attributes of the players).

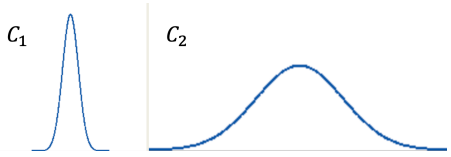
$$C_{Def} = \frac{\Sigma_i (W_i . {P^i}_{Def})}{\Sigma_i W_i}$$

The weights depend on the Overall ratings of the players:

- Overall $>$ 85: Weight = 20
- 80 $<$ Overall $<$ 85: Weight = 16
- 75 $<$ Overall $<$ 80: Weight = 8
- 70 $<$ Overall $<$ 75: Weight = 2
- Overall $<$ 70: Weight = 1

This distribution ensures that players with high overall rating(who play regularly) affect the mean value more. Aside from this, there was also an implicit weight given to the amount of time a player spent at the club(in years). This was achieved as the weighted mean was taken over all 5 years of data and the same player in different year was treated as a different player (Messi-2019 and Messi-2020 are treated as different players). This lead to the player being counted in the mean as many times as he was at the club, thus affecting the weighted mean more.

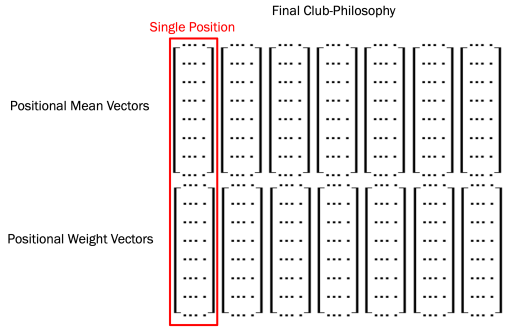
Now we have 7 vectors for each Club representing the average player that club wants in each of the 7 positions. However, we haven't included the aspect of **How** much the club wants a particular attribute to be the way it is. Consider the following example: There are two clubs C_1, C_2 whose Defenders have the following distribution of the **HeadingAccuracy** attribute as shown in the Figure.



We can clearly say that C_1 has more emphasis on what kind of **HeadingAccuracy** its defenders have as compared to C_2. This means that even from the essential attributes in a position, clubs have different preferences for different attributes. We model this by giving weights to these essential attributes. The weights of a position P's attributes in a club C are reciprocals of the standard deviation of top 20 players (4 first team players x 5 years) in P playing for C over the 5 years. For example, let the set of top 20 players be S for a Club C. Then

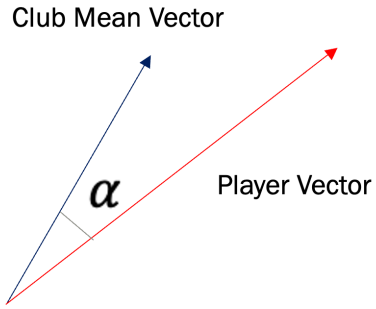
$$W_{Def}(C) = \begin{bmatrix} StdDev_S(HeadingAccuracy)^{-1} \\ StdDev_S(ShortPassing)^{-1} \\ StdDev_S(LongPassing)^{-1} \\ StdDev_S(BallControl)^{-1} \\ StdDev_S(Reactions)^{-1} \\ StdDev_S(Aggression)^{-1} \\ StdDev_S(Intereptions)^{-1} \\ StdDev_S(Composure)^{-1} \\ StdDev_S(StandingTackle)^{-1} \\ StdDev_S(SlidingTackle)^{-1} \end{bmatrix}$$

So, now we have a total of 14 vectors to represent a Club. 2 for each position(A Weighted Mean vector and a Weight vector) as shown in the Figure.



B. Working

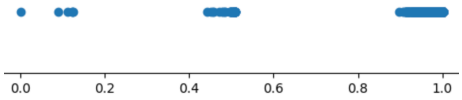
Now we use above defined philosophy to find players for the club. Lets talk about a particular position now. We want to recommend players to a club in that particular position(Lets say Defender). We take the C_{Def} vector of the Club and plot it in 10-dimensional space. We also take the player vectors P^i_{Def} of all the Defenders that are available and plot them too in the same space. As shown in the Figure, our immediate goal is to recommend players with least angle from C_{Def} vector irrespective of the vector magnitudes.



Immediate solution would be Weighted Cosine Similarity (CS) with weights coming from the W_{Def} vector. And then rank the players according to this Similarity rating.

$$CS^i = \frac{\Sigma_{j=0}^{j=10}(W_{Def}[j].C_{Def}[j].P^i_{Def}[j])}{\sqrt{\Sigma_j(W_{Def}[j].C_{Def}[j]^2)}\sqrt{\Sigma_j(W_{Def}[j].P^i_{Def}[j]^2)}}$$

Although theoretically seeming to work, practically the discriminatory power of Cosine Similarity was very less as shown in distribution plot in the Figure.



We decided to use Pearson's Correlation Coefficient instead. But this posed another problem, Pearson's Coefficient starting taking the vector magnitudes as factors and made high overall players similar to each other irrespective of angular distance.

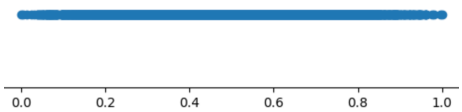
To solve this we decided to normalize our player vectors and Club vector which removed the magnitude of vectors from the whole setting. Now Pearson's Coefficient was forced to find similarity on basis of angular distance. We used Weighted Pearson's Correlation Coefficient.

$$\Delta C = C_{Def} - Mean$$

$$\Delta P^i = P^i_{Def} - Mean$$

$$sim^i = \frac{\Sigma_{j=0}^{j=10}(W_{Def}[j].\Delta C[j].\Delta P^i[j])}{\sqrt{\Sigma_j(W_{Def}[j].\Delta C[j]^2)}\sqrt{\Sigma_j(W_{Def}[j].\Delta P^i[j]^2)}}$$

This similarity index worked wonders as shown in the distribution in the Figure.



We still cannot directly rank players according to this similarity index for recommendations. This index only showed how similar the player is to the Club's philosophy. But what we want to recommend were Players who are **Similar** and **Good** too. So we need to introduce a balance between Similarity

and Strength of the player. But using the overall rating of the player as a measure of strength was not enough.

Different clubs would have a different perspective on how strong a player is. So we define the strength (m) metric to be the weighted magnitude of the player's vector where the weights are from the club's weight vectors. Thus,

$$m^i = \sqrt{\frac{\Sigma_{j=0}^{j=10}(W_{Def}[j].P_{Def}^i[j]^2)}{\Sigma_{j=0}^{j=10}(W_{Def}[j])}}$$

Now, we balance the strength m^i and similarity sim^i with a linear relation to find the rating r^i:

$$r^i = 0.63 * sim^i + 0.37 * m^i$$

The recommendation list is the list of players sorted in descending order using this rating.

C. Analysis

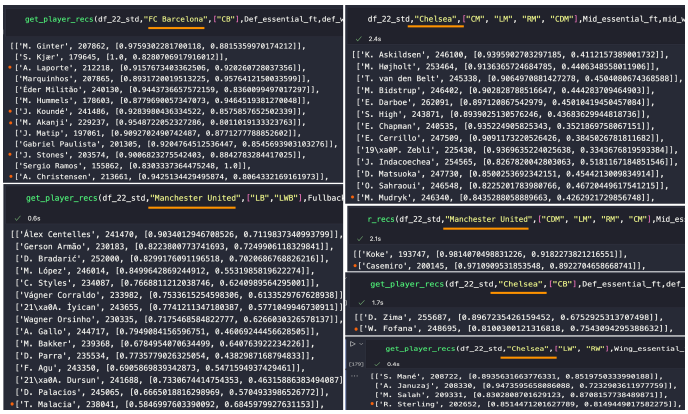
We relate our results with ground truth transfers that took place at the start of 2022-2023 season because our data has been trained on the data before that. We have added more such results according to Prof. Raghuram's feedback. More analysis is in the further sections. For testing purpose we needed prominent clubs which have been doing a team build-up recently. FC Barcelona, Manchester United and Chelsea FC are good candidates.

Note: Age-Range and Max-Cost range are user adjustable in our model. So the results have been calculated by keeping the Age-Range within +-2 of the player's age and Max-Cost rounded off to the least multiple of 10 greater than player's cost. (If Player age is 22 and Cost is 4.5, then the parameters are: Age-Range = 20-24 and Max-Cost = 10). Some other parameter such as queried positions etc. were also tuned(For example if the player is a Left-Back then only Left-Back players were queried for ratings, not Right-Backs).

Name	Recommended To	Joined	Position	Approach-4 Rank
Mudryk	Chelsea	Chelsea	Midfield	13
Fofana	Chelsea	Chelsea	Defender	2
Felix	Chelsea	Chelsea	Att. Midfield	336
Cucurella	Chelsea	Chelsea	Midfield	14
Sterling	Chelsea	Chelsea	Winger	4
Enzo	Chelsea	Chelsea	Midfield	7
Kounde	FCB	FCB	Defender	7
Lewandowski	FCB	FCB	Striker	3
Christensen	FCB	FCB	Defender	13
Akanji	FCB	Manchester City	Defender	8
Stones	FCB	Manchester City	Defender	11
Laporte	FCB	Manchester City	Defender	3
Antony	Manchester Utd	Manchester Utd	Att. Midfield	14
Casemiro	Manchester Utd	Manchester Utd	Midfield	2
Malacia	Manchester Utd	Manchester Utd	FullBack	15

We were surprised that our model performed well even for Chelsea which led us to conclude that they didn't really have as chaotic, haphazard and impulsive a transfer season as it seemed. Another pattern recognised by our model was the similar play style of FC Barcelona and Manchester City and the history of player trades between both clubs. FC Barcelona was recommended 3 of Manchester City's top Defenders. Following table shows numerous players who actually joined the mentioned Clubs this season. We can also see that Felix was ranked 336 in our recommendations for Chelsea, this actually reflects how he was signed impulsively overnight just minutes before the deadline of the transfer window.

Here are a few screenshots to validate the tables.



Considering the outstanding results of this model, this is our **Flagship Model** for the project. Although, it does have one prominent disadvantage: **Lack of Serendipity**. The model only recommends what the club thinks would work because it already works. But we have to ask the question: Is it necessary that players who aren't as similar to club philosophy cannot fit well or even better? For example, Erling Haaland's play-style is in complete contrast what Manchester City like, but in his debut season with the Club, he has broken the 27 year old Premiere League Goals Record in a Single Season with multiple games still left to play. We try to answer this question in next approach.

XI. Approach 5: XGBoost and Neural Networks

We extend the ideas and philosophy established in the previous approach to recommend players on the basis of how good they already are and how much they are predicted to improve if they join the given club.

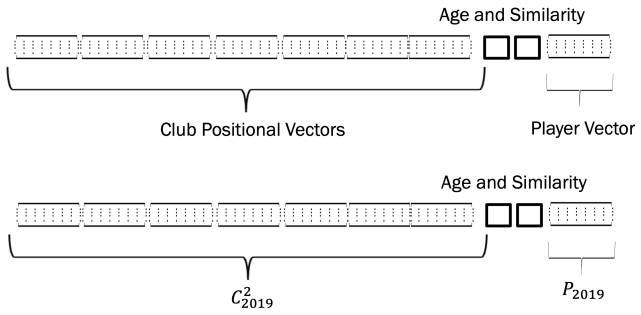
A. Feature Engineering

The Club Philosophy(all 7 weighted mean vectors), Player philosophy (Player Vector), Player Age and Similarity(Same as approach-4) between Club and Player, all together represent the inputs to the Regression Models. We had to engineer the target column, which is the improvement of the player on joining the Club(Change in Overall Rating over the debut season).

B. Working

We considered every year to be a new transfer for the player(even if he remains at the same club). This was done assuming that the same club in a different year is still a different club and similarly for the players. Lets take an example, a player P went from club C^1 to club C^2 at the

start of year 2020. C^1 and C^2 might be the same Club. Player data for a year is represented as P_{year} and Club data as C_{year}. Given Figure shows training inputs and targets.



Why put all the 7 Club vectors in the input features? This was done so that the Regression Model learns patterns that show inter-positional effects on performance. For example, A striker with great heading ability could score a lot if the midfielders have great crossing ability even though striker isn't very similar to club's striker mold.

C. Analysis

We had to train the XGBoost and Neural Network on Kaggle. This poised a problem as we couldn't train Neural Networks/XGBoost for all 7 positions. For predicting also we had to first export prediction dataset to kaggle, then import the predicted values to local machines and then predict. We did the whole process for Defence position for 2 clubs - "Manchester United" and "Manchester City" considering the time constraint, and extracted as much ground-truth coherence as we could. As we established Approach-4 to be working really good with the ground-truth, we also used it as a benchmark along with the ground truth. Apart from that, MSE values of Neural Network and XGBoost also helped. XGBoost also helped us validate our philosophy by using Feature importance scores from XGBoost.

1) Neural Network: When trained on the above defined dataset, we got the following results from a Neural Network which was run for 200 Epochs using L2 regularization. The target value ranged from -3.5 to 7.5 and MSE was about 0.68 as seen in the Figure.

Optimizer	Mean Squared Error
Adam	0.6826
Adagrad	0.6823
Adadelta	0.6840

Epochs:200, Adam optimizer

Learning rate	Mean Squared Error
0.001	0.6826
0.01	0.6853
0.1	0.6824

2) XGBoost: The XGBoost Regression gave us a very similar MSE of 0.685. We used 1000 estimators with a depth of 6 and a learning rate of 0.01. In a bid to improve the performance we queried the Importance scores of the features from XGBoost as seen in the Figure.

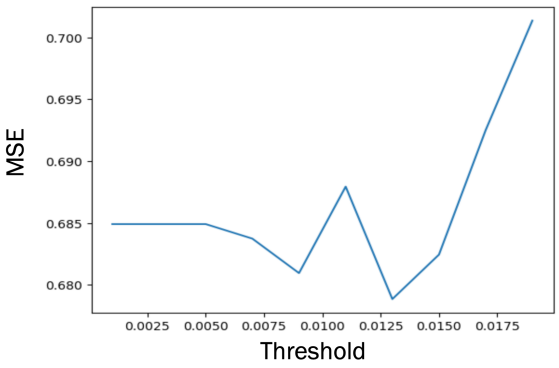
```
print(my_model.feature_importances_)
```

```
[0.03273191 0.00851815 0.00635185 0.0078481  0.0088491  0.01566508
 0.01592314 0.01828243 0.03617587 0.0132319  0.01224892 0.00798366
 0.00894529 0.00907722 0.00808141 0.01350628 0.01073165 0.01127564
 0.00801901 0.00924683 0.00880632 0.01266967 0.01039656 0.00947976
 0.01312917 0.00758138 0.00847808 0.01772967 0.01019354 0.02454421
 0.01179754 0.01323241 0.01150341 0.01590569 0.01068776 0.01739481
 0.01001501 0.01229732 0.01069494 0.00916138 0.00795054 0.0108042
 0.0118127  0.00856235 0.01347495 0.01145019 0.01121532 0.01107345
 0.01118613 0.00749227 0.01145793 0.0099155  0.00909001 0.01236769
 0.0111692  0.01094123 0.01500657 0.01052115 0.01233654 0.01015758
 0.01821454 0.01053189 0.01652169 0.00988168 0.01187515 0.01600802
 0.01362472 0.01186936 0.01428742 0.00905751 0.01449527 0.01531938
 0.0092067  0.01076354 0.0090439  0.01095743 0.01149364 0.01012931
 0.01012998 0.01159441 0.01350665 0.01511029]
```

The importance scores list also told us that Age and Similarity(In red) were actually important features affecting the improvement of a player.

Now, by setting a threshold over the importance scores we trained another XGBoost model with features whose importance scores were more than the threshold. We found threshold 0.013 to give the least MSE of 0.67 as seen in the Figure.

```
import matplotlib.pyplot as plt
plt.plot(thresholds, y)
plt.show()
```



3) Recommendation Analysis: As mentioned we have recommendation for Manchester United and Manchester City for the Defence Position. We take ground truth as benchmark for comparison. As both the clubs have not signed many defenders recently, we rely on media articles which show that the club and mentioned players were or are in talks for a transfer. These media articles are quite reliable as almost every football transfer always comes in the media articles first, surprise transfers are extremely rare. And the articles also tell us that the club is interested in the player. Following table shows the results:

Name	Recommended To	Reported to Join (Media reports)	Position	Approach-5 Rank
Kounde	Manchester Utd	Manchester Utd	Defender	1
Tapsoba	Manchester Utd	Manchester Utd	Defender	2
Bastoni	Manchester Utd	Manchester Utd	Defender	5
Botman	Manchester City	Manchester City	Defender	5
Ndicka	Manchester City	Manchester City	Defender	2

Media article links in order are:

- Kounde
- Tapsoba
- Bastoni
- Botman
- Ndicka

Now, we took Approach-4 as a benchmark as it has been observed to perform well in previous sections.

Name	Recommended To	Position	Approach-5 Rank	Approach-4 Rank
Diveev	Manchester City	Defender	12	2
Vanheusden	Manchester City	Defender	8	1
Markovic	Manchester City	Defender	23	11
Parrela	Manchester City	Defender	1	8
Kabak	Manchester City	Defender	7	4
Carlos	Manchester Utd	Defender	3	3
Chiamuloira	Manchester Utd	Defender	7	9
Virissimo	Manchester Utd	Defender	10	7
Bensabini	Manchester Utd	Defender	12	24

We notice that the recommendation ranks do not vary a lot between approach-4 and approach-5. This means similarity(approach-4 concept) plays a pivotal role in improvement of a player. Another observation is that for example for Diveev and Markovic, their approach-4 and approach-5 ranks differ by about 10 places. We think this is because they are from the not-so-strong leagues of Greece and Russia whereas Manchester City is a top club in the English league. This parity of level might be playing a role in this.
This model also showed two other real-world patterns,

- Players who are already very good, had lesser improvement. This is trivial as the room for improvement would be less. Our model showed this pattern and thus we had to draw a balance between the current strength of the player and their predicted improvement. This was done through a linear relation.
- When a highly rated player goes to a new club which has a different philosophy and style of playing, they usually struggle with game time and performance in their debut season. This dip in overall was modeled by our approach. Very highly rated players did have negative improvements and were removed from the recommendation list by our algorithm.

XII. Result Analysis

According to Prof. Raghuram's feedback, we use this section to compare approaches 3 and 4 (approach 5 has only been implemented for defenders). This was done by taking the ground truth as benchmark. We again take the Clubs FC Barcelona, Chelsea and Manchester United.

Name	Recommended To	Joined	Position	Approach-3 Rank	Approach-4 Rank
Mudryk	Chelsea	Chelsea	Midfield	1566	13
Fofana	Chelsea	Chelsea	Defender	301	2
Felix	Chelsea	Chelsea	Att. Midfield	1518	336
Cucurella	Chelsea	Chelsea	Midfield	231	14
Sterling	Chelsea	Chelsea	Winger	614	4
Enzo	Chelsea	Chelsea	Midfield	1114	7
Kounde	FCB	FCB	Defender	514	7
Lewandowski	FCB	FCB	Striker	461	3
Christensen	FCB	FCB	Defender	2551	13
Akanji	FCB	Manchester City	Defender	883	8
Stones	FCB	Manchester City	Defender	1721	11
Laporte	FCB	Manchester City	Defender	1049	3
Antony	Manchester Utd	Manchester Utd	Att. Midfield	1539	14
Casemiro	Manchester Utd	Manchester Utd	Midfield	1445	2
Malacia	Manchester Utd	Manchester Utd	FullBack	996	15

We can observe that Approach-3 is giving really bad results. The reason for this is that approach-3 defined a very basic philosophy which didn't really have any effect on the improvement. So the neural network learnt to use just the age and overall of the player to predict improvement. This also resulted in all clubs getting very similar recommendations. On the other hand the Club philosophy defined in approach-4 plays a pivotal role and does very good when compared to ground truth. This shows the importance of having a sophisticated club and player philosophy. The work done in this project was one of the first of its kind. This kind of mathematical representation of Football has not been implemented before to the best of our knowledge(There have been very basic attempts online). Nevertheless, there have been other theoretical works which try to model the Football world using more advanced concepts such as passing rate and expected goals.

XIII. Summary

A. Social Impact

- Inclusion and Diversity: Football clubs have historically relied on traditional scouting methods that often lead to a narrow focus on certain regions, ethnicity, or physical characteristics. This is a dialogue from a scout in the movie **Moneyball** which highlights scouting bias: "Clean-cut, good face. Yeah, good jaw. Got an ugly girlfriend… means no confidence." These traits are completely unrelated to a player's performance on the pitch.
- Fairness and meritocracy: A recommendation system can help remove bias and subjectivity from the player selection process by relying on objective data and performance metrics.
- Opportunity for players: The use of a recommendation system can provide more opportunities for talented but underrepresented players to showcase their skills and get discovered by clubs.
- Efficiency and cost savings: Traditional scouting process can be time-consuming, expensive, and ineffective. By,

using a recommendation system, clubs can save time and resources by focusing on a targeted list of recommended players who are more likely to meet their needs and criteria.
- Recommendation Systems can help reduce the number of bad transfers at all levels of Football. A bad transfer is catastrophic for everyone, especially the player. A player's career is essentially over after a bad transfer, current club doesn't need them, other clubs don't seek them.

B. AI-ML Ideas and Concepts

- Recommendation paradigms : Collaborative and Content Based recommendation systems.
- ML models : Linear regression, XGBoost, Decision Tree, KNN etc
- Deep Learning : Working with neural networks
- NLP : Working with Bert
- Data Pre-processing
- Statistical Similarity Measures: Cosine Similarity, Pearson's Correlation Coefficient
- Feature Engineering

C. Novelty

- A sophisticated yet simple and efficient definition of club philosophy that completely describes the type of players in the club and the type of players that will fit well into the club. (First of its kind)
- Using 5 different models and analysing the shortcomings and advantages of each.
- A user friendly front-end that even a layman can use to get recommendations.
- Using Important feature selection to improve XGBoost perfomance
- Use of an NLP transformer like Bert that uses the description of players to provide recommendations.

D. Dataset Creation and Synthesis

Given only basic player data we needed to define club philosophy and positional philosophy of a club.

- Year on Year Improvement: maximum 5 values for each player.
- Club philosophy: 140 values per club(14 vectors).
- Player philosophy: 10 values per player
- Player-Club similarity
- Average club rating
- Average club potential

E. Future Scope

- Features in text form containing information about the player's playstyle could help boost performance of NLP and transformer-based models.
- A "Money Ball" like system where we recommend high rated players who have been overlooked by other clubs. The club using the system will be able to get good layers at a lower price.
- A club recommendation for players, in case a player has multiple offers , using the improvement based model.
- Defining a Club's Overall Improvement metric to be used as the0 target in order to model how much a Club improves if a player joins.
- Refine metrics and features.
- Since the FIFA data is sequential year-wise data, RNNs can be used.

REFERENCES

[1] https://towardsdatascience.com/using-machine-learning-to-identify-high-value-football-transfer-targets-d4151a7ffcac
[2] https://www.kaggle.com/code/ekrembayar/fifa-data-analysis-visualization
[3] Palash Goyal,Anna Sapienza, Emilio Ferrara, Recommending Teammates with Deep Neural Networks
[4] https://www.kaggle.com/datasets/bryanb/fifa-player-stats-database
[5] https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/
[6] https://www.kaggle.com/code/blessontomjoseph/feature-eng-for-player-recommendation-pes-data/notebook