

Report
FIFA players recommendation system
Cristina Acevedo

1. Introduction

FIFA ultimate team is one of the most popular play station games. My brother has played this game for years and has seen the game evolve from a basic soccer match simulator into what it is today. Nowadays, there is a lot more than soccer in this game, there are different ways to play, including managing a team. In this play mode, players must make strategic decisions to create a team that contains players who perform well and who have good chemistry with other team players. However, this team is constrained by the limitations of the market. Players have values that vary based on their attributes and the market's offer and demand. This is why I have decided to create a recommendation algorithm that can suggest players who are similar to a chosen one. Therefore, when the users are considering which players to buy, they can check some options that might fit their team and that were not previously considered.

2. Data

The data for this project was obtained from [Kaggle](#). It contains different attributes of 4169 players in FIFA19. Although the dataset is outdated (there are new FIFA versions every year), it is a good place to start examining the data and getting to know what each attribute means. Here is a preview of the dataset.

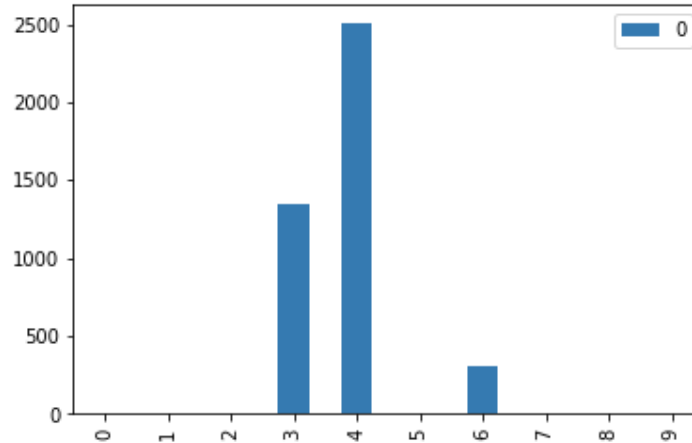
| Unnamed: 0 | ID | Name | Age | Photo | Nationality | Flag | Overall | Potential | Club | Club Logo | Value | Wage | Special | Preferred Foot | |
|---------------------|----|--------|-------------------|-------|---|-----------|---|-----------|------|---------------------|---|---------|---------|----------------|-------|
| 0 | 0 | 158023 | L. Messi | 31 | https://cdn.sofifa.org/players/4/19/158023.png | Argentina | https://cdn.sofifa.org/flags/52.png | 94 | 94 | FC Barcelona | https://cdn.sofifa.org/teams/2/light/241.png | €110.5M | €565K | 2202.0 | Left |
| 1 | 1 | 20801 | Cristiano Ronaldo | 33 | https://cdn.sofifa.org/players/4/19/20801.png | Portugal | https://cdn.sofifa.org/flags/38.png | 94 | 94 | Juventus | https://cdn.sofifa.org/teams/2/light/45.png | €77M | €405K | 2228.0 | Right |
| 2 | 2 | 190871 | Neymar Jr | 26 | https://cdn.sofifa.org/players/4/19/190871.png | Brazil | https://cdn.sofifa.org/flags/54.png | 92 | 93 | Paris Saint-Germain | https://cdn.sofifa.org/teams/2/light/73.png | €118.5M | €290K | 2143.0 | Right |
| 3 | 3 | 193080 | De Gea | 27 | https://cdn.sofifa.org/players/4/19/193080.png | Spain | https://cdn.sofifa.org/flags/45.png | 91 | 93 | Manchester United | https://cdn.sofifa.org/teams/2/light/11.png | €72M | €260K | 1471.0 | Right |
| 4 | 4 | 192985 | K. De Bruyne | 27 | https://cdn.sofifa.org/players/4/19/192985.png | Belgium | https://cdn.sofifa.org/flags/7.png | 91 | 92 | Manchester City | https://cdn.sofifa.org/teams/2/light/10.png | €102M | €355K | 2281.0 | Right |
| 5 rows x 89 columns | | | | | | | | | | | | | | | |

As it is shown, there are 89 different columns that contain different kinds of data. Ideally, all of this data will have to be transformed into quantitative measures in order to determine the similarity between players. Some of these features contain NaNs or empty fields, which is something to consider once developing the model.

3. Methodology

First, the fields that did not provide useful information for the model were eliminated, as they could introduce noise. These fields were 'Unnamed: 0', 'ID', 'Flag', 'Club Logo' and 'Real Face'. Then, I noticed that the variable called 'Preferred

'Foot' had only two unique values, therefore the values were replaced with either 1 or 0 to have binarized field. Another technique to turn a qualitative field into a quantitative one is one-hot encoding. In this case, the 'Position' was codified through the use of this technique and new columns corresponding to the possible values were added. The 'Body Type' was a very special case, as there were 10 unique values with the following distribution:



As it is shown, only three of these values had a significant frequency in the database, which meant that only these were going to be useful when looking for similar players. These three body types were codified as with values from 0-2, as they had an order. The 'Work Rate' field was split as it contained two different types of information (according to my brother) and then the values were codified with values from 0-2 as they also presented an order. Then, the column 'Loaned from' was transformed into a binary column called loaned, as the team from where the player was loaned was not relevant. 'Nationality' and 'Club' were also turned into quantitative fields by one-hot encoding. 'Value', 'Wage', 'Weight', 'Height' and 'Release Clause' were transformed into a numerical class identifying the numbers in the field and deleting other characters. This is the description of the data frame at this point:

```
df.describe()
```

| | Age | Overall | Potential | Value | Wage | Special | Preferred Foot | International Reputation | Weak Foot | Skill Moves |
|-------|-------------|-------------|-------------|-------------|-------------|-------------|----------------|--------------------------|-------------|-------------|
| count | 4169.000000 | 4169.000000 | 4169.000000 | 4109.000000 | 4110.000000 | 4168.000000 | 4168.000000 | 4168.000000 | 4168.000000 | 4168.000000 |
| mean | 27.251379 | 75.376589 | 77.646198 | 23.155658 | 30.640633 | 1810.418666 | 0.245921 | 1.449856 | 3.133877 | 2.786708 |
| std | 4.017433 | 3.613685 | 4.538256 | 104.159034 | 38.979662 | 243.498931 | 0.430684 | 0.696510 | 0.709782 | 0.874043 |
| min | 17.000000 | 71.000000 | 71.000000 | 1.000000 | 1.000000 | 918.000000 | 0.000000 | 1.000000 | 1.000000 | 1.000000 |
| 25% | 24.000000 | 73.000000 | 74.000000 | 3.700000 | 11.000000 | 1740.000000 | 0.000000 | 1.000000 | 3.000000 | 2.000000 |
| 50% | 27.000000 | 74.000000 | 77.000000 | 6.000000 | 20.000000 | 1866.000000 | 0.000000 | 1.000000 | 3.000000 | 3.000000 |
| 75% | 30.000000 | 77.000000 | 81.000000 | 10.000000 | 35.000000 | 1963.000000 | 0.000000 | 2.000000 | 4.000000 | 3.000000 |
| max | 41.000000 | 94.000000 | 95.000000 | 975.000000 | 565.000000 | 2346.000000 | 1.000000 | 5.000000 | 5.000000 | 5.000000 |

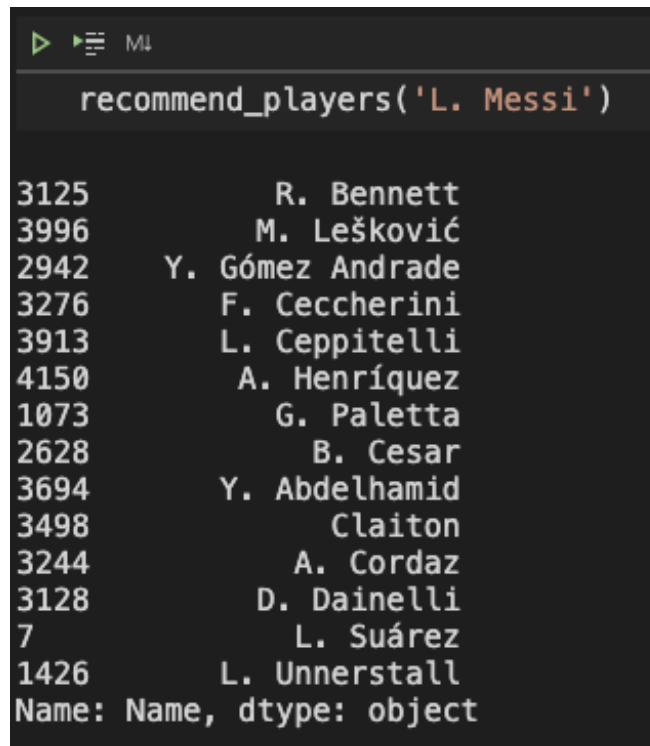
8 rows x 11 columns

Finally, I saved the progress at this point and dropped some other columns that might not have useful information for the model ('Name', 'Joined' and 'Contract Valid Until'). After this was done, I started creating the model by scaling all the variables

between 0 and 1 using the MinMaxScaler. Then, I replaced all of the NaNs with the average for that field and proceeded to create the similarity matrix using `pairwise_distances`. The last step consisted on mapping the players' names to the matrix's rows and creating a function that will find the 14 higher values for the similarity and return the names of these players.

4. Results

A recommender function was created based on the selected and processed features from the players. This function takes a players' name as the input and then searches for other players using a similarity matrix. From the different kinds of recommender algorithms, a content-based algorithm was chosen. This kind of algorithm does not require information from the user which makes it ideal for this application. It is important to consider that the method gives each one of the features the same importance and modifying these weights could potentially provide better recommendations. Here is an example of how the developed function works:



```
recommend_players('L. Messi')
```

| | |
|------|------------------|
| 3125 | R. Bennett |
| 3996 | M. Lešković |
| 2942 | Y. Gómez Andrade |
| 3276 | F. Ceccherini |
| 3913 | L. Ceppitelli |
| 4150 | A. Henríquez |
| 1073 | G. Paletta |
| 2628 | B. Cesar |
| 3694 | Y. Abdelhamid |
| 3498 | Claiton |
| 3244 | A. Cordaz |
| 3128 | D. Dainelli |
| 7 | L. Suárez |
| 1426 | L. Unnerstall |

Name: Name, dtype: object

5. Discussion

The results of the algorithm seem to make sense, since the recommended players do have some similarity to the input player. There are several aspects that could be improved in order to create a better experience while using this system. First, I would like to create an interface where user could feel more comfortable. I think it would be a good idea to create a web app using shiny or Flask which will make everything more user friendly and provide more options to the user as, for example, the use of filters or getting other features besides the players' name as an output. Additionally, it would be

necessary to replace the data used to develop this model. The data available in Kaggle is from 2019, but FIFA releases a new game every year with frequent updates that include new players or player cards every few weeks. This is why it is also necessary to develop a system to constantly update the database. An option would be using an API such as <https://futdb.app/> to get information from the players, although this might mean a new data processing algorithm will be necessary as well.

6. Conclusion

The developed algorithm is able to provide a list of similar players from a single player's name input, which was the goal of this project. The data processing of different features allowed the use of different kinds of variables to develop the algorithm, instead of only using quantitative values. There are still several ways in which this project can be improved, and the results show a clear path to follow in order to make it easier to use and provide better suggestions (I am still waiting on my brother's approval).