

Scouting App Report

Problem characterization in the application domain

This abstraction level describes specific issues of the application domain and end-users involved, such as the problem to solve, user demands, and datasets.

The domain of the application is football analysis. More specifically, it is focused on scouting, where specialists work on finding the players that best fit the requirements they are looking for, and they also need to have tools that provide them with information to make proper decisions. The questions and situations that may arise are:

- **Q1:** Which player is better between these two? Scouts need to be able to compare players based on multiple possible attributes.
- **Q2:** Which players are similar to each other? Scouts need to find groups of players who have similar characteristics.
- **Q3:** Is there any correlation or trend in some of the players' characteristics? Which players stand out from the crowd? Scouts need to find out which features are correlated and who are outliers in their positions.
- **Q4:** From which countries does each league sign the players? Is there a country where you find a higher number of forwards, defenders...? Scouts need to find the countries in which to look for future talent.
- **Q5:** What is the approximate salary we will have to pay a player? How much will I have to pay to transfer a player with specific characteristics? Scouts need indicators to sense spending on players.
- **Q6:** Scouts also need to explore the dataset simply and interactively to perform easy queries such as sorting and filtering.

The datasets we can find on football are very diverse: historical data on teams, results and match statistics, player data, etc. To solve scouting problems, we need data segregated to each player's level and focused on their defining characteristics.

Data and task abstractions

At this second level, we make abstractions of the specific tasks and data types, mapping the particular issues of the domain to a generic representation.

Data abstractions

The main dataset is of table type. A table is a set of items structured in rows and columns. Each row represents an item and each column represents an attribute. It contains items that correspond to each of the players and attributes, which would define the characteristics of each of the items. It is a multivariable table where each item is identified by a key (player's name). The key is unique and exclusive to each item, there are no two players with the same name.

- Key: Name
- Attributes: Club, League, Nation, Position, Age, Value, Salary, Foot, Pace, Shooting, Passing, Dribbling, Defending, Physic and Rating.

The type of data used is intrinsically related to the characterization of the problem and the objectives, so the type of data must be adapted to the user's problem to be solved. For questions Q1, Q3, Q5 and Q6 a table data type approach was followed. For Q4, the scout needs to find the countries in which to look for future talents, so to achieve this goal, a geometry data type was used. Maps are a type of geometry data that relate some attributes of the items with geographic information. For Q2 cluster data type was used. Clusters store or encompass a finite set of items.

Task abstractions

We define the tasks abstractions based on the set of questions that we collect from the analysts. The three levels of actions that could be considered user goals or needs are: why is the visualization being used? (**use**), what kind of search is performed? (**search**) and what kind of query? (**query**).

- For Q1, the **use** is to *discover* and *verify* because scouts are discovering and verifying which player is better in relation to their interests and prior assumptions, the **search** type is to *lookup* because two previously known selected players are compared, the **query** type is obviously to *compare*, and the **targets** are the *features* (of the players).
- For Q2, the **use** is to *produce* and *derive* new data. It is necessary to perform clustering to identify the groups of players and create this new cluster variable that generates a new visual encoding, also 2 variables coming from PCA are created. The **search** type is to *explore* because the scout does not know where to find the similarities between players (they do not know the number of clusters), and they do not know precisely what to search. The **query** type is to *summarize* because the scout is browsing all targets. The **targets** are the *features*, the players themselves and the clusters or groups (with their characteristics).
- For Q3, the **use** is to *consume* information, to *discover* new knowledge to *generate* or *verify* some hypothesis. The scout may have in mind players similar to a certain player or discover new players who are also similar to that reference player. He may also discover correlations between player attributes based on pre-established ideas or novel correlations that will guide future player scouting. The **search** type is *lookup* because the target is known and also the location. The scout knows both, what and where to look. The **query** is to *compare* multiple players in relation to their characteristics. To answer this question, we are working with all the data items. The **targets** are the *items* and also the *features*. We can find outliers that would correspond to those players who stand out either positively or negatively in the two attributes to be compared. On the other hand, trends and correlations between attributes can also be observed to help the scout in his search for players with certain qualities.
- For Q4, the **use** is to *discover* and generate hypotheses about the distribution of the players' nations. The search type is to *locate* countries where to find new players, but the user does not know where to find them. The **query** type is to summarize because

they look at all the possible targets, and we are aggregating them by the player's nation. The **target** is outliers in the number of players of some countries.

- For Q5, the **use** is to *present* information. The **search** type is to *locate* an indication of what to offer a player, since the user knows what they are looking for but not where to look for it. The **query** type is to *summarize*, since we return an average salary or value for the scout to base his decision on. The **target** is *features*, where we want to find the individual value that gives the value or salary indication based on a certain age or rating.
- For Q6, the **use** is to enjoy or discover information to generate or verify hypotheses. The **search** type is to *explore* since the scout does not know where or what to look. The **target** is *features* of the players.

Interaction and visual encoding

In this third level, we define 6 different idioms, following the same scheme of the original questions and abstract tasks, by specifying how the different design options are built to create and manipulate the visualizations. This “how” is expressed with the actions **encode**, **manipulate**, **facet** and **reduce**, and their specific *internal choices*.

Idiom 1 (Compare players)

- **Encode**: As **visual marks**, we use points to represent the value of the quantitative attribute (Pace, Shooting, Passing, Dribbling, Defending or Physic) and a line connecting those points to represent a player. As **channels**, we consider the hue to differentiate both players. And the **arrange** choice is to **express** the features using a **radar chart**. This decision is based on the fact that the main task is not to internally compare the player's characteristics (for which a linear layout would be better), but we want to compare him with another player. In order to be able to compare correctly, the configuration of the characteristics has been executed considering that the characteristics that define a good striker should appear together, those that define a defender should appear together, and so on.
- **Manipulate**: we decide to apply the option to **change** the view to hide the visualization of one of the players if the scout wants to visualize only one.
- **Facet**: the option to show both players' characteristics in the same view is **superimpose** because visually is the fastest way to compare them.
- **Reduce**: In order to control the items displayed, the choice is a **filter** to allow the scout to choose the two players they want.

Idiom 2 (Similar players)

- **Encode**: The data used in this visualization are the 8 attributes of a player in which we perform a PCA to be capable of printing them in 2D. So the data that we plot are 2 quantitative variables (attributes) and 1 categorical variable (cluster). As **visual marks**, we use points to represent a player in the 2 dimensions (x-axis contains dimension 1 and y-axis includes dimension 2). As **channels**, we use hue and shape to differentiate better each cluster, if we only use hue or shape, some limit points between clusters will be more difficult to distinguish. This scatterplot is used to see

the level of overlapping of each cluster and visualize how well they are defined. Taking this into account, the scalability of this plot is up to a few hundred points.

- **Manipulate:** The scout can **select** if he wants to see the cluster summary or a list with the players represented in the visualization, and explore them in detail. This is done in a table format and not in the visualization itself to better explore the players' attributes.
- **Facet:** All the clusters are **superimposed** to display all of them in the graphic.
- **Reduce:** We include a **filter** for the position, the leagues and the number of clusters. It has no sense to perform clustering with defenders and forwards since they are totally different players. With the leagues' filter, the scout can choose a set of leagues. And the number of clusters can vary depending on the interpretation of the chart.

Idiom 3 (Stats Correlation)

- **Encode:** The data used in this visualization is composed of 2 quantitative variables and one categorical variable. Spatial **arrangement** designed choices were done. The values of the quantitative variables are expressed with horizontal and vertical spatial position (x and y-axis) as they are the most important attributes of this visualization. We have made this decision based on the principle of expressiveness, so the most important attributes are encoded in the most relevant **channels** (spatial). We have used the color channel to encode the categorical variable of the foot (left or right) using red for left-footed players and blue for right footed players. Players are encoded as point marks and the name of the player is displayed above the point for easy identification. It is a **scatterplot** design with a scalability of dozens of items. The presence of the names facilitates their interpretation but limits the scalability of the visualization.
- **Manipulate:** The scout can **select** the players in the view (point marks) to display a more detailed description of the player and his skills. This description will appear in a pop-up box.
- **Facet:** The left-footed players are **superimposed** to the right-footed players to display both in the graphic.
- **Reduce:** In the left part the scout can **filter** the League, Position, Characteristic 1 and Characteristic 2, to reduce the number of players displayed and do a more specific search.

Idiom 4 (Nations)

- **Encode:** The data used in this visualization aggregates the number of players by each nationality. The data type is geometry since the nations are represented in a map. We use a **choropleth map** representing each country to arrange this spatial data. As a channel, to represent the number of players we use a **diverging color map** that combines hue with saturation or luminance establishing a scale. The diverging color map lets the scout distinguish more clearly nations with almost no players and countries with lots of them, this can be done with a sequential color map also, but this contrast is visually less noticeable. The map is used to locate regions and see grouping patterns of closer countries. We include another visualization for a more detailed view of the distribution of players among the most important countries

(top 8). This is done through a radial axis representation (**pie chart**) in which we use the hue to differentiate the countries and the **area** to watch its proportion over the total. This can be done through a bar chart that is more accurate, but we consider that there are many items for bars. Also, to compensate for this lack of accuracy, we include the % inside the area of each country.

- **Reduce:** The scout can **filter** and choose the league and the position. It is necessary to make this division between leagues to observe the different patterns between them, and the scout may want to look at specific positions.

Idiom 5 (Offer Indication)

- **Encode:** As **visual marks** the data has been encoded in a barchart which arranges the data aligned with the x-axis, either rating or age, to see a difference between the values represented on the x-axis for the value of the y-axis, either average value or average salary. This has been chosen so the scout can not only find a good indication of what he should offer a player but also if he maybe should try and find a player with slightly different characteristics who might be a better price-quality ratio.
- **Manipulate:** For the interaction with the graph the scout is able to zoom in and/or hover over each bar. Zooming shows smaller differences which might not appear when fully zoomed out and hovering gives the exact values of each bar. Both of these manipulations give the scout the ability to distinguish small differences between groups.

Idiom 6 (Database)

- **Encode:** The visualization for exploring the database is an **interactive data table**, so no decisions about marks, channels or spatial arrangement are needed.
- **Reduce:** Multiple types of **filters** are introduced to filter players in their most important attributes. Also, the possibility of ordering attributes is included.

Algorithm implementation

This final level intends to implement the previous encodings and interactions techniques effectively. The app has been implemented in Shiny, and it is structured again in 6 different tabs corresponding to the defined idioms in the level above.

Tab Compare players


In this tab, you can see the visualization with the radar chart that allows you to compare the attributes of the players. Thanks to the filters implemented on the side, it is possible to search for the desired players by typing them. A photograph of each player is also shown with indicative data on the player (age, value, position and nationality).


Select Players

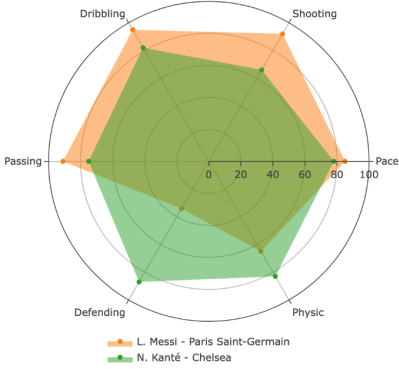
Player 1:
L. Messi - Paris Saint-Germain

Player 2:
N. Kanté - Chelsea

After clicking on the selector you should write the name of the player.
To visualize / hide the stats of a player click his name in the legend.

L. Messi

AGE: 34
VALUE: € 78 M
POSITION: RW
NATION: Argentina

N. Kanté

AGE: 30
VALUE: € 100 M
POSITION: CDM
NATION: France



Legend:
L. Messi - Paris Saint-Germain
N. Kanté - Chelsea

Tab Similar players

This tab shows the clusters' chart that allows you to see the groups of similar players and how they overlap in the two dimensions resulting from PCA. At the bottom, you can choose to view a summary of the clusters or browse the players within each cluster and find players that are similar to each other. In addition, the scout can apply filters to choose the position, leagues and number of clusters he wants.

Clustering

Select Position:
Defender

Select Leagues:
English Premier League, French Ligue 1, German 1. Bundes

Number of Clusters:
3

The visualisation helps to choose the appropriate or desired number of clusters.

Kmeans (3 clusters)

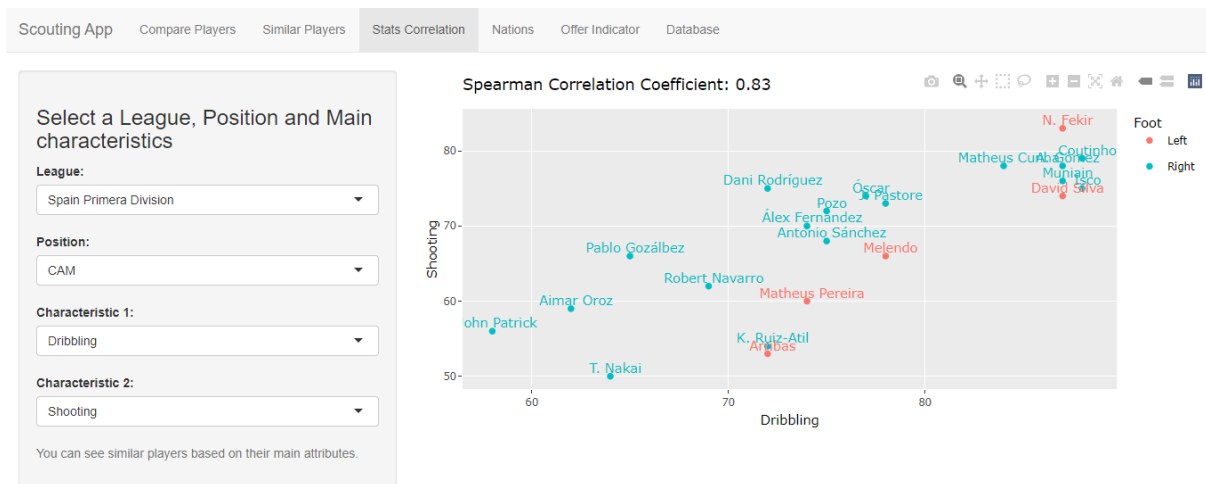


Legend: cluster 1 (red), 2 (green), 3 (blue)

Summary **Players**

Cluster	Number of Players	Value (€M)	Salary (€k)	Age	Pace	Shooting	Passing	Dribbling	Defending	Physic	Main Position
1	367	14.05	44.74	26	76	56	69	72	73	72	LB
2	320	9.33	33.61	26	61	39	57	60	75	75	CB
3	240	1.35	6.93	21	62	33	46	52	62	63	CB

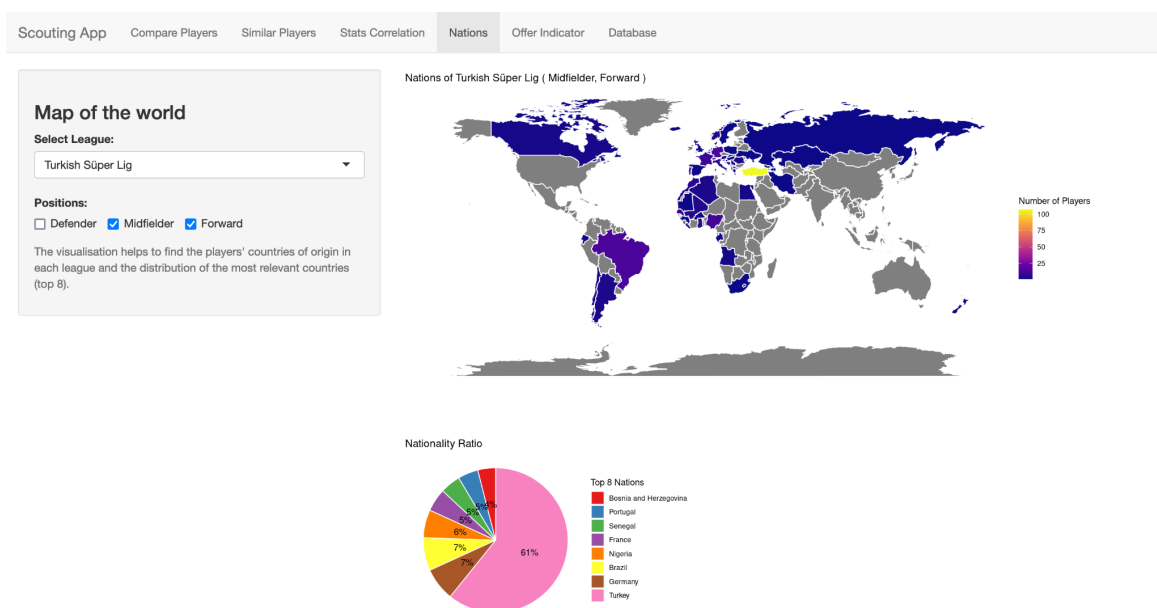
Tab Stats Correlation



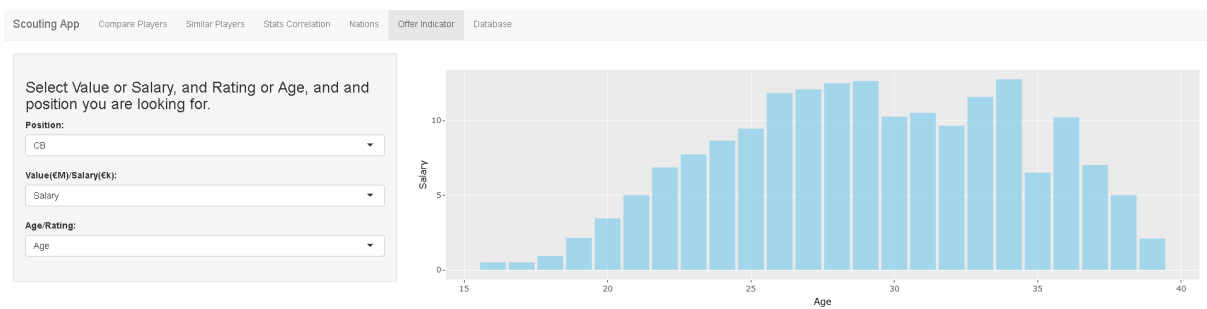
With this visualization, the scout can find the players with the best skills based on two characteristics that act as a filter. The search can be filtered according to League and Position. Left-footed and right-footed players are differentiated by the color of the point marks so that the scout can more easily find the player that best suits his requirements. Finally, the correlation between the two filtered characteristics is shown. The correlation data help the scout to find related attributes that may dictate the future player search.

Tab Nations

In this tab, you can see the visualisation of the map, where you can observe the countries of origin of the players. The scout can locate which countries have a greater number of players or which ones do not have any. You can also see a pie chart, which allows you to see what percentage of the players in a league are from each nation (only the top 8 countries). The filters included are for selecting the leagues and the positions.



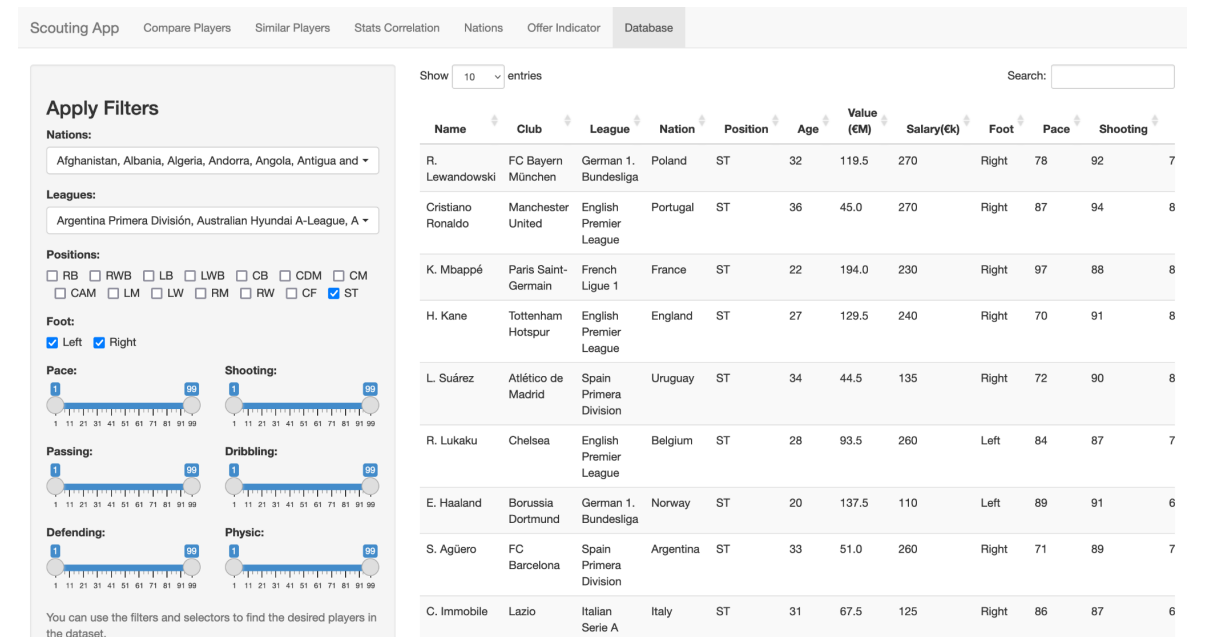
Tab Offer indicator



With this visualization the scout can get an indication on what the transfer fee and salary of the player he is interested in should be based on the average players similar to this player make, and in some cases, the scout could change their scouting criteria based on the graph shown. For example, when a player is a couple of years younger the scout’s offer could be half of that of the player they initially scouted. But of course, this is to the discretion of the scout and the knowledge of the field they possess.

The visualization itself shows the average salary or value based on the selected filter by the user for the age or rating of players, also based on the selected filter of the user. This can be done for each position which requires the user to select a position.

Tab Database



This tab shows the database used for the visualisations. It can be explored by using the filters on the side, sorting the table, or searching directly in each column.

How to run the app

To run the app, you can use the command `runApp("scouting-app")` inside the R interpreter or execute it inside the `app.R` in R Studio.

Or just click on the following link where is hosted our app:

<https://kasperlange.shinyapps.io/scouting-app/>

The dependencies are:

- tidyverse
- plotly
- factoextra
- shiny
- shinydashboard
- shinyWidgets
- maps