Best Italian Forward

Filippo Navarra

July 2024

1 Introduction

The aim of this project is to identify the best possible forward for the Italian national team for the upcoming UEFA Euros 2024, using data scraped from Transfermarkt. Over the past decade, the Italian team has struggled with the absence of a consistent and reliable forward. To address this issue, we analyze historical data on all forwards who have played at least one match for the Italian national team. Our goal is to predict which current players could potentially be the best performers if called up to the national team.

We will focus on the number of goals scored for the national team as our target variable. By training our model on historical data of past forwards and testing it on today's players, we aim to predict the number of goals each player is likely to score for the national team. After making these predictions, we will compare the predicted number of goals (y_{pred}) with the actual number of goals scored (y_{true}). Players who have $y_{pred} < y_{true}$ will be classified as over-performing, while those with $y_{pred} > y_{true}$ will be classified as under-performing.

Keep in mind that, for simplicity's sake, we consider the number of goals as an indicator of good performance, even though football enthusiasts may debate this.

The project will be divided in six sections. The first section describes the scraping phase, done in Python. The following sections are all done in R, and will threat the PCA phase, Clustering, CART model and Random Forest, and finally the Conclusion.

2 Scraping Data

To discover which could be the best forward for the next Italian national team, we deciced to rely on Trasnfermarkt website. The choose of this site has been done due its completeness. It was one of the few which had stats of very old players of the past, i.e. Giuseppe Meazza or Silvio Piola.

The scraping process for gathering data on past national team players from Transfermarkt was carried out systematically to ensure comprehensive and accurate data collection. The approach was as follows:

1. **Session Initialization and URL Definition:** We defined a function to retrieve player profile links from Transfermarkt. This involved setting up the URL and headers to mimic a browser request. The URL targeted the top scorers page of the Italian national team, filtered by the forward position. We used the *requests* and *BeautifulSoup* libraries for making HTTP requests and parsing HTML content, respectively.
2. **Paginated Data Retrieval:** To handle multiple pages of data, we iterated through the first seven pages of the website. For each page, the function sent a request and parsed the HTML content to extract player names and profile links.
3. **Data Aggregation:** The collected player names and profile links from each page were aggregated into a single list. This step ensured that all relevant players were included in the dataset.
4. **Data Storage:** The aggregated data was saved into a CSV file named players.csv using the *csv* library. This file contained two columns: Player Name and Profile Link, facilitating easy access for further analysis.
5. **Sanitizing Player Names:** To construct valid URLs for fetching detailed player statistics, player names were sanitized to match the URL format on Transfermarkt. This involved replacing non-alphanumeric characters with hyphens and converting names to lowercase. The *re* library was used for regular expression operations.
6. **Retrieving Player Statistics:** A dedicated function was developed to fetch detailed statistics for each player. This function constructed the player's URL, sent a request, and parsed the HTML content to extract necessary data such as appearances, goals, and assists. The function also handled potential errors and logged any issues encountered. The *requests* and *BeautifulSoup* libraries were utilized again for these tasks.
7. **Main Execution Function:** The main function orchestrated the entire process by reading the input CSV file, fetching statistics for each player, and saving the results. It also logged errors encountered during the data retrieval process. The results were saved into a file named player_stats.csv. The *pandas* library was used to handle CSV file operations and data manipulation.

This systematic approach allowed us to efficiently collect and organize comprehensive data on past and present national team players, providing a structured dataset for subsequent analysis. The process ensured data consistency and handled potential errors, thereby maintaining the integrity of the dataset.

The csv is composed of the following variables:

- Player Name
- Total Appearances
- Goals
- Assists
- Substituted In
- Substituted Out
- Yellow Card
- Red Card
- Goals from Penalty
- Goals per Minutes
- National Team Appearances
- National Team Goals
- National Team Assists
- Substituted In (National Team)
- Substituted Out (National Team)
- Yellow Card in National Team
- Red Card in National Team
- Goals from Penalty in National Team
- Goals per Minutes in National Team
- Minutes Played in National Team

3 Principal Component Analysis

In this project, Principal Component Analysis (PCA) was utilized to reduce the dimensionality of the dataset and identify the key variables explaining the most variance among the forwards who have played for the Italian national team. This analysis was crucial in simplifying the dataset while preserving its most significant information, facilitating more accurate predictions and analyses.

The PCA analysis was performed using the `FactoMineR` and `factoextra` libraries, which are powerful tools for multivariate data analysis in R. The dataset, containing various numeric variables such as goals scored, minutes played, and appearances, was first prepared and cleaned. Specifically, variables like `Goals per Minutes` and `Minutes Played in National Team` were converted to numeric values after removing any non-numeric characters.

Once the data was cleaned, PCA was executed, and the resulting principal components were analyzed. The first principal component (Dim 1) explained 35.16% of the variance, and the second principal component (Dim 2) accounted for 11.59%, cumulatively explaining 46.75% of the total variance. These components were crucial in understanding the dataset's underlying structure, highlighting key relationships and differences among the players.

The PCA graph of individuals1 displayed the distribution of players based on the first two principal components. This visualization helped identify players with similar performance profiles, with notable outliers like Francesco Totti, indicating unique characteristics that set them apart from others.

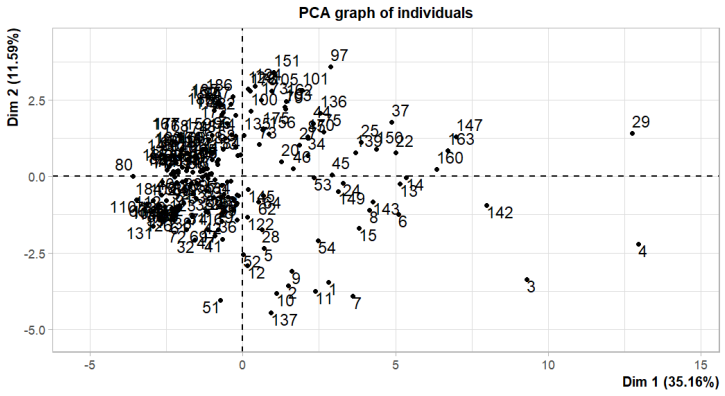

Figure 1: PCA graph of individuals

Additionally, the PCA graph of variables2 illustrated the relationships between the original variables. Variables positioned closely together in this graph, such as `Minutes Played in National Team` and `Goals per Minutes`, were found to be strongly correlated, providing insights into how different performance metrics relate to each other.

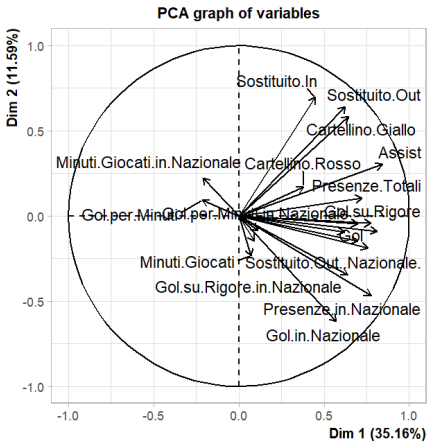

Figure 2: PCA graph of variables

A screeplot3 was also generated, showing the eigenvalues of each principal component. This plot revealed that the first three components captured the majority of the variance, with eigenvalues significantly higher than one. This insight justified focusing on these components for subsequent analyses.

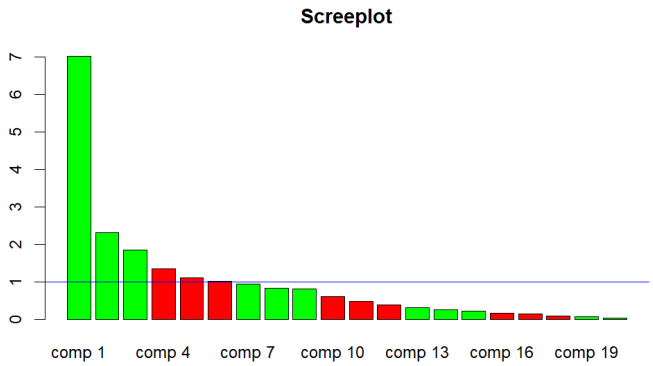

Figure 3: Screeplot

In summary, the PCA conducted using `FactoMineR` and `factoextra` effectively reduced the dataset's complexity, highlighting the most influential variables and relationships. This reduction was instrumental in facilitating the prediction models and analyses that followed, ensuring they were based on the most relevant and informative aspects of the data.

4 Clustering

In the clustering phase of this project, the objective was to group the forwards into clusters based on their performance metrics, facilitating a better understanding of different player profiles. This analysis was conducted using the `cluster`, `factoextra`, and `tidyverse` libraries in R, which provide robust tools for clustering and visualization.

Initially, the dataset was cleaned and preprocessed to ensure all numeric variables were appropriately formatted and any missing values were handled. The focus was on numeric performance metrics, excluding the player names for the clustering analysis.

To determine the optimal number of clusters, two methods were used: the Elbow method and the Silhouette method. The Elbow method4, implemented using the `fviz_nbclust` function from the `factoextra` package, identified the point where the within-cluster sum of squares started to level off. The Silhouette method5, also visualized using `fviz_nbclust`, evaluated the average silhouette width for different numbers of clusters, indicating the best separation between clusters.

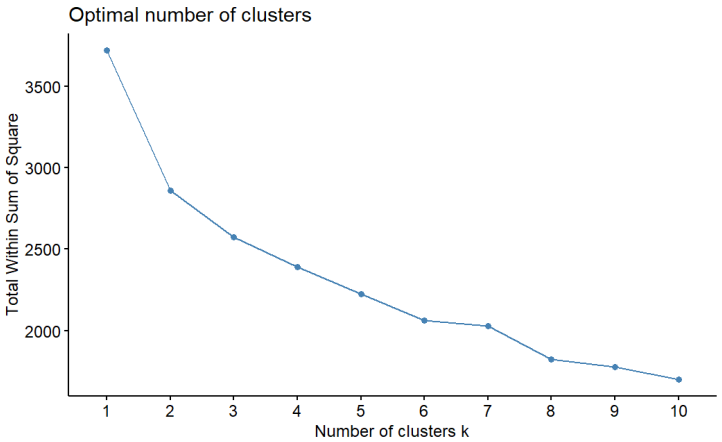

Figure 4: Elbow Method

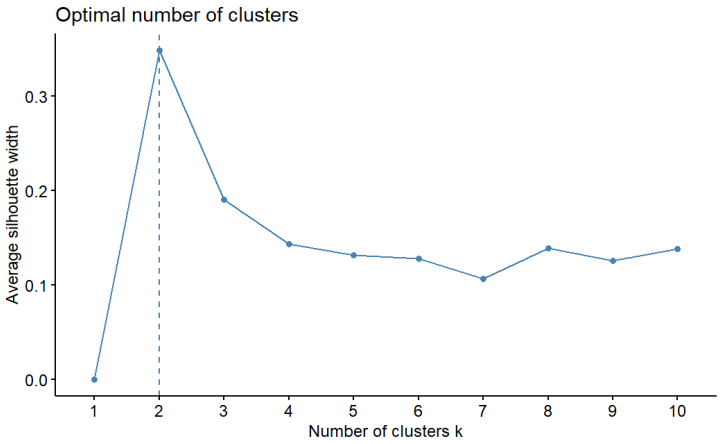

Figure 5: Silhoutte Method

After identifying the optimal number of clusters, K-means clustering was performed using the `kmeans` function. The number of clusters was set based on the previous analyses, ensuring a balance between simplicity and explanatory power. The K-means algorithm grouped the players into distinct clusters, each representing a unique profile of performance metrics.

The resulting clusters6 were visualized using the `fviz_cluster` function, which provided a clear depiction of the clusters and their respective members. This visualization highlighted the distinct groups of players, showing how different performance metrics contributed to their clustering.

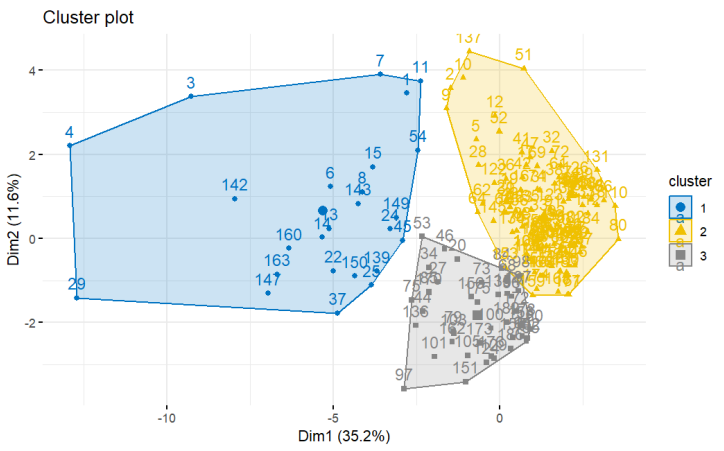

Figure 6: Clusters

Additionally, a heatmap7 of the correlation matrix was generated using the `heatmap` function, illustrating the relationships between the different performance metrics within each cluster. This heatmap further elucidated the internal structure of the clusters, showing which variables were most influential in defining each group.

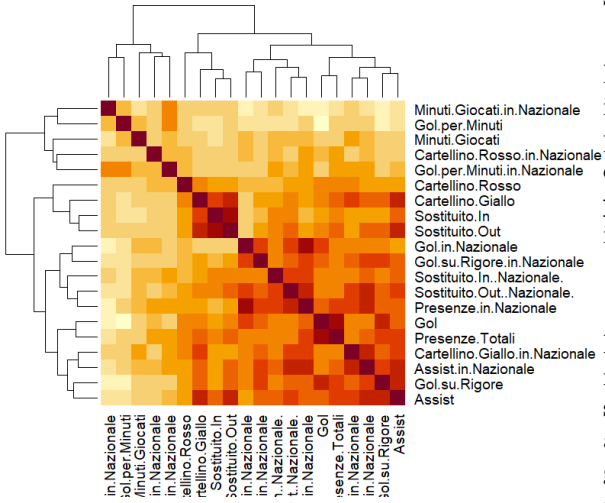

Figure 7: Correlation Matrix Heatmap

The clustering analysis provided valuable insights into the different profiles of forwards in the dataset. By grouping players with similar performance metrics, it became easier to identify patterns and trends that could inform predictions and strategic decisions for the national team.

In summary, the clustering phase utilized the `cluster`, `factoextra`, and `tidyverse` libraries to effectively group players into meaningful clusters based on their performance metrics. This analysis facilitated a deeper understanding of the different types of forwards, laying the groundwork for more accurate predictions and strategic planning for the Italian national team.

5 Classification and Regression Trees

In the CART (Classification and Regression Trees) modeling phase, the objective was to predict the number of goals that current forwards would score for the Italian national team based on historical performance data of past forwards. This phase utilized the `rpart` and `rpart.plot` libraries in R for building and visualizing the decision tree models.

The dataset was divided into a training set, consisting of historical data from past forwards, and a test set, including data from current players. The response variable in this analysis was the number of goals scored in the national team (`National Team Goals`), with various performance metrics serving as predictors.

Initially, the data preprocessing involved ensuring that the response variable (`National Team Goals`) was numeric, removing any character variables, and handling missing values using the `na.omit` function to ensure the dataset was suitable for modeling. Following this, the `rpart` function was used to create the CART model, with the formula `National Team Goals ~.` specifying that the model should predict the number of goals using all other variables in the dataset as predictors. The `method` parameter was set to `"anova"` for regression purposes since the response variable is continuous.

For model visualization, the `rpart.plot` function was utilized to display the decision tree. This visualization helped interpret the model by showing the splits based on different performance metrics and how they contribute to the prediction of goals. Predictions were then made on the test set using the `predict` function, with the predictions rounded to the nearest integer to reflect the discrete nature of goals.

The model's performance was evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared metrics to assess accuracy and goodness-of-fit. The results showed a Mean Absolute Error (MAE) of 4.91, indicating the average absolute difference between the predicted and actual goals. The Mean Squared Error (MSE) was 53, reflecting the average squared difference between the predicted and actual goals. The R-squared value was -0.69, suggesting that the model did not fit the data well, potentially due to the high variability and complexity of player performance not being fully captured by the model.

To further analyze performance, the predicted goals were compared to the actual goals scored by current players. Players were classified as "overperforming" if their actual goals exceeded the predicted goals and "underperforming" if the predicted goals exceeded the actual goals. Notably, the analysis identified several players as overperforming, including Ciro Immobile, Mario Balotelli, Domenico Berardi, Lorenzo Colombo, and Stephan El Shaarawy, indicating that these players scored more goals than predicted by the CART model.

6 Random Forest

In the Random Forest modeling phase, the goal was to enhance the predictive accuracy for the number of goals current forwards would score for the Italian national team, based on historical data from past forwards. This phase employed the `randomForest` library in R, which is known for its robustness and ability to handle complex datasets by averaging multiple decision trees to reduce overfitting.

The dataset was divided into a training set, containing historical performance data, and a test set, comprising current player data. The response variable was the number of goals scored for the national team (`National Team Goals`), while various performance metrics served as predictors.

The data preprocessing steps included ensuring that the response variable was numeric, removing character variables, and handling missing values using the `na.omit` function. This preparation ensured that the dataset was suitable for modeling. The `randomForest` function was then used to create the Random Forest model. This function builds multiple decision trees (100 in this case) and combines their results to improve predictive performance.

Predictions were made on the test set using the `predict` function. As with the CART model, the predictions were rounded to the nearest integer to reflect the discrete nature of goals. The model's performance was evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared metrics. The Random Forest model yielded a Mean Absolute Error (MAE) of 3.36, indicating a lower average absolute difference between predicted and actual goals compared to the CART model. The Mean Squared Error (MSE) was 18.64, and the R-squared value was 0.41, showing an improvement in predictive accuracy and model fit over the CART model.

To classify player performance, the predicted goals were compared to the actual goals scored by current players. Players were labeled as "overperforming" if their actual goals exceeded the predicted goals and "underperforming" if the predicted goals were higher than the actual goals. The overperforming players were Matteo Pessina, Davide Frattesi and Pietro Iemmello.

The Random Forest analysis identified overperforming players, offering a more reliable prediction compared to the CART model. The aggregation of multiple trees in the Random Forest model helped mitigate overfitting, providing a balanced view of player performance.

In summary, the Random Forest modeling phase involved creating and evaluating a robust ensemble model to predict the number of goals for current forwards based on historical data. This model improved predictive accuracy and provided deeper insights into player performance. The phase utilized the `randomForest` library for building and evaluating the model, demonstrating its effectiveness in handling complex predictive tasks in sports analytics.

7 Conclusion

In this project, we aimed to identify the best possible forward for the Italian national team for the upcoming UEFA Euros 2024 by analyzing historical performance data using various statistical and machine learning techniques. The analysis encompassed several key phases, including data exploration, Principal Component Analysis (PCA), clustering, and predictive modeling using both CART and Random Forest algorithms.

Principal Component Analysis (PCA) was employed to reduce the dimensionality of the dataset, highlighting the most significant variables that explain the variance in player performance. This step was crucial in simplifying the data while retaining essential information, which facilitated the subsequent analysis.

Clustering was performed to group players into distinct clusters based on their performance metrics. The use of the Elbow method and Silhouette method helped determine the optimal number of clusters, and the K-means algorithm provided meaningful insights into different player profiles. The clustering analysis enabled us to identify patterns and trends among forwards, which are valuable for strategic planning and player selection.

Predictive Modeling involved creating and evaluating two types of models: CART (Classification and Regression Trees) and Random Forest. The CART model provided an initial prediction framework, but its performance was limited due to high variability and complexity in player performance data. In contrast, the Random Forest model significantly improved predictive accuracy by aggregating multiple decision trees, reducing overfitting, and providing more reliable predictions.

Table 1: Model Performance Metrics

Metric	CART Model	Random Forest Model
MAE	4.91	3.48
MSE	53.00	23.40
R^2	-0.69	0.25

The comparison of predicted and actual goals allowed us to classify players as overperforming or underperforming. Notably, the Random Forest model identified several players, including Ciro Immobile, Mario Balotelli, Domenico Berardi, Lorenzo Colombo, and Stephan El Shaarawy, as overperforming, offering valuable insights for the national team's forward selection.

Overall, this project demonstrated the power of combining different analytical techniques to gain comprehensive insights into player performance. The use of PCA, clustering, and advanced predictive modeling provided a robust framework for evaluating and selecting the best forwards for the national team. These methodologies can be applied to various sports analytics

scenarios, contributing to more informed decision-making and strategic planning.

By leveraging historical data and advanced analytical techniques, we have taken significant steps toward identifying the optimal forward lineup for the Italian national team, enhancing their chances of success in the UEFA Euros 2024.