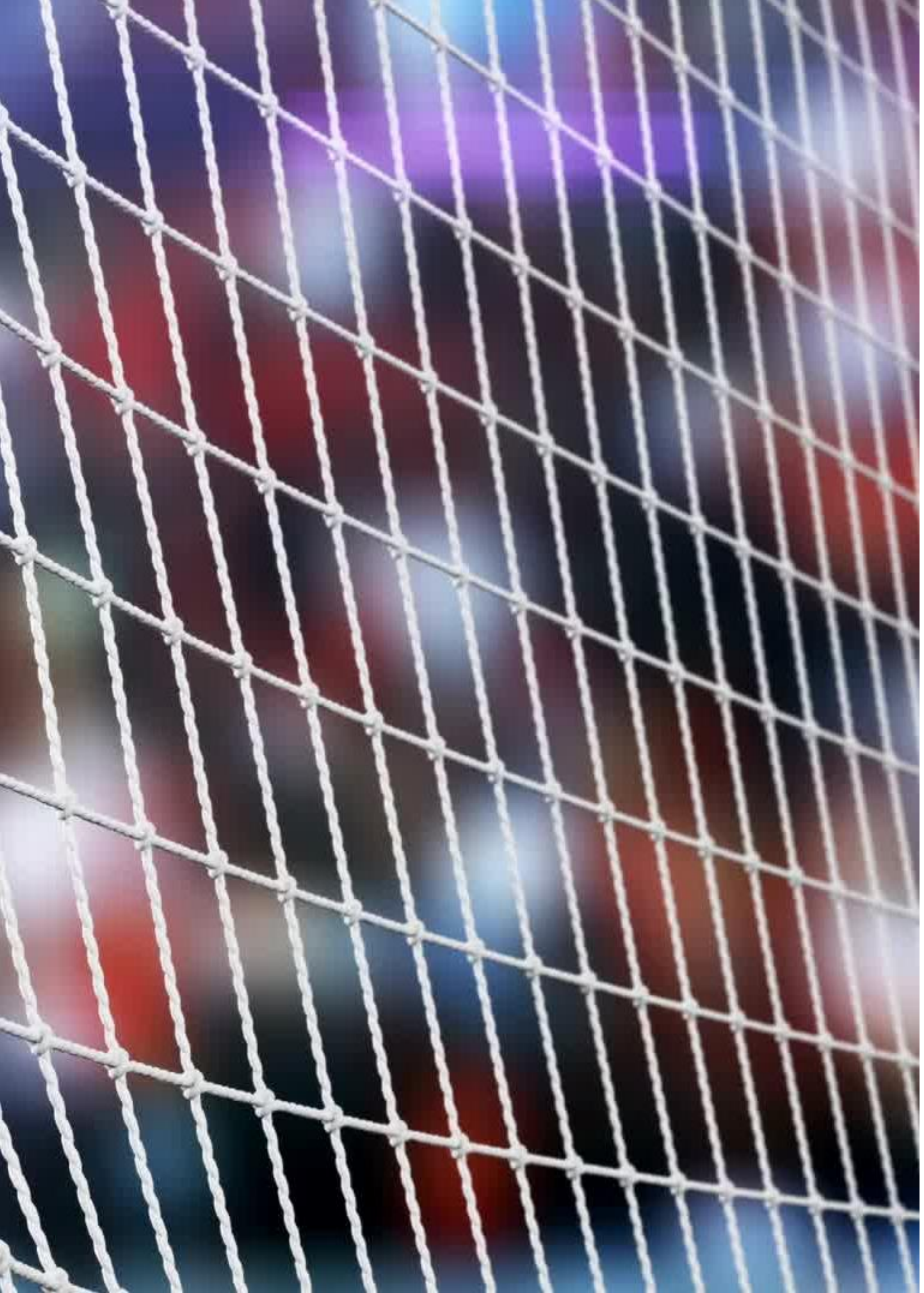


SOCCER RATINGS

Data mining and machine learning

Giuseppe Martino

INTRODUCTION

- *Soccer Ratings* is an application intended to evaluate players performances in a match, assigning them a **rating** from 1 to 5 entering as input some statistical data
- The application is also able to help a coach suggesting a team's **best formation** for the next match based on the ratings given to each player in the last 5 matches.
- Newspapers and sports magazines are used to evaluate the performances of footballers after a match assigning them a rating
- Different criteria may be used by each evaluator to decide whether a footballer has played well or not

DATASET

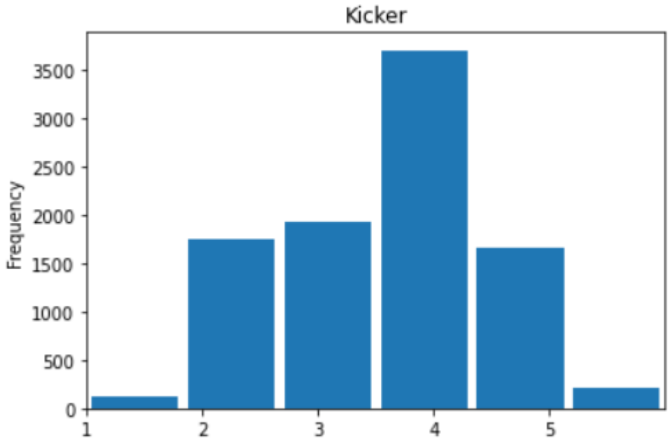
- 50652 instances and 63 attributes
- 789 different matches across 4 different competitions between 2016 and 2018
- Ratings are taken from 6 different sport magazines and specialized websites

competition	date	match ▲	team	pos	pos_role	player	rater	original_rating	goals	assists	shots_ontarget	shots_offtarget	shotsblocked
Bundesliga 2017-18	04/11/2017	Augsburg - Bayer Leverkusen, 1 - 1	Bayer Leverkusen	MF	AML	Julian Brandt	Kicker	4.0	0	0	0	0	0
Bundesliga 2017-18	04/11/2017	Augsburg - Bayer Leverkusen, 1 - 1	Bayer Leverkusen	MF	AML	Julian Brandt	WhoScored	6.71	0	0	0	0	0
Bundesliga 2017-18	04/11/2017	Augsburg - Bayer Leverkusen, 1 - 1	Bayer Leverkusen	MF	AML	Julian Brandt	Bild	4.0	0	0	0	0	0
Bundesliga 2017-18	04/11/2017	Augsburg - Bayer Leverkusen, 1 - 1	Bayer Leverkusen	MF	AMC	Kai Havertz	Kicker	3.0	0	0	0	1	0
Bundesliga 2017-18	04/11/2017	Augsburg - Bayer Leverkusen, 1 - 1	Bayer Leverkusen	MF	AMC	Kai Havertz	WhoScored	7.98	0	0	0	1	0

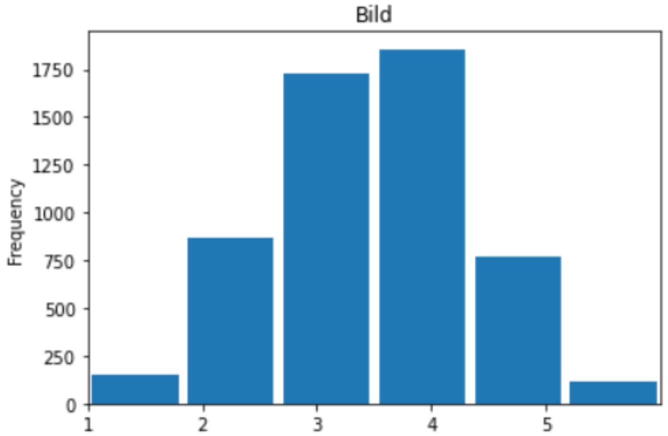
• • •

⋮

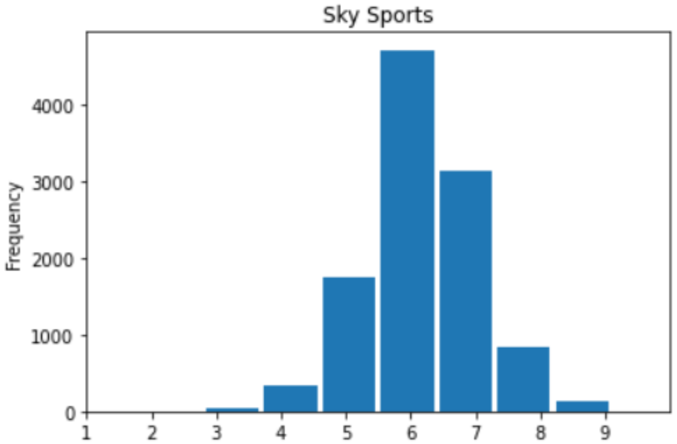
Ratings distribution for each magazine



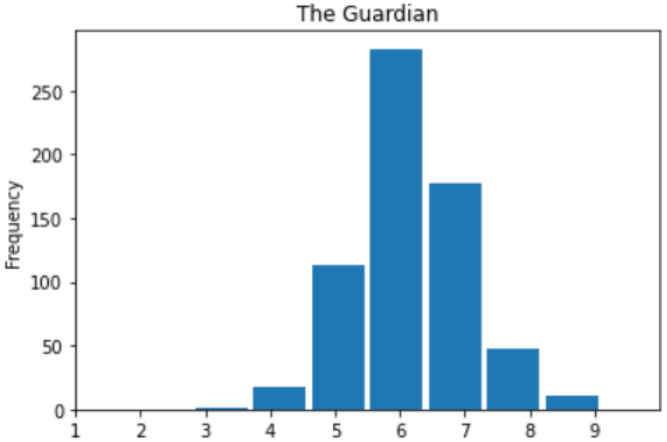
- 1 to 6 scale in 0.5-size steps
- descending order of goodness of performance
- rating is discrete



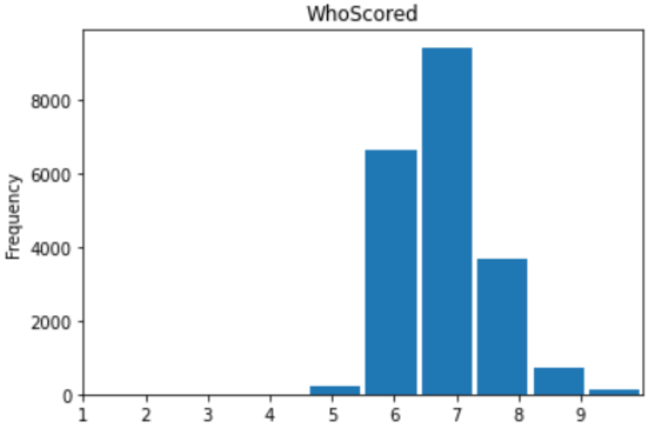
- 1 to 6 scale
- descending order of goodness of performance
- rating is discrete



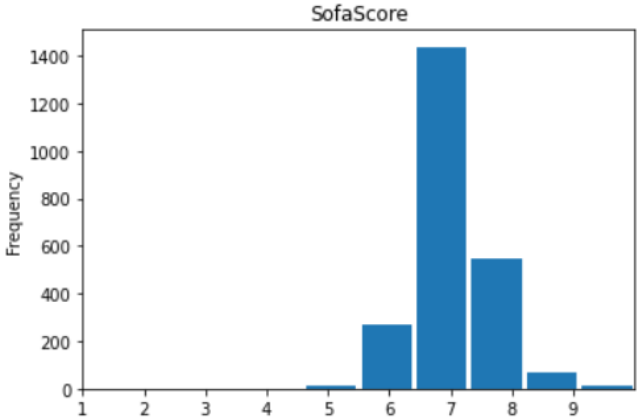
- 1 to 10 scale
- ascending order of goodness of performance
- rating is discrete



- 1 to 10 scale
- ascending order of goodness of performance
- rating is discrete



- 1 to 10 scale
- ascending order of goodness of performance
- rating is continuous

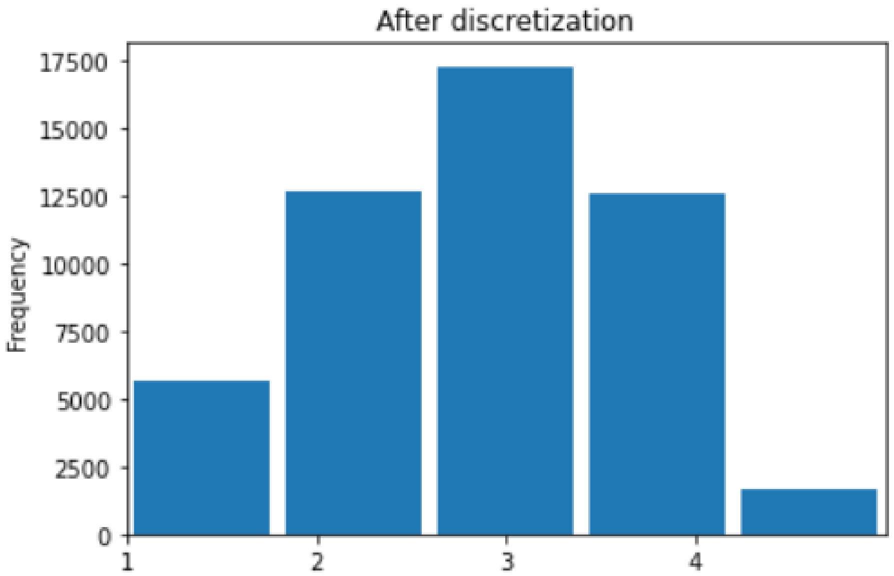


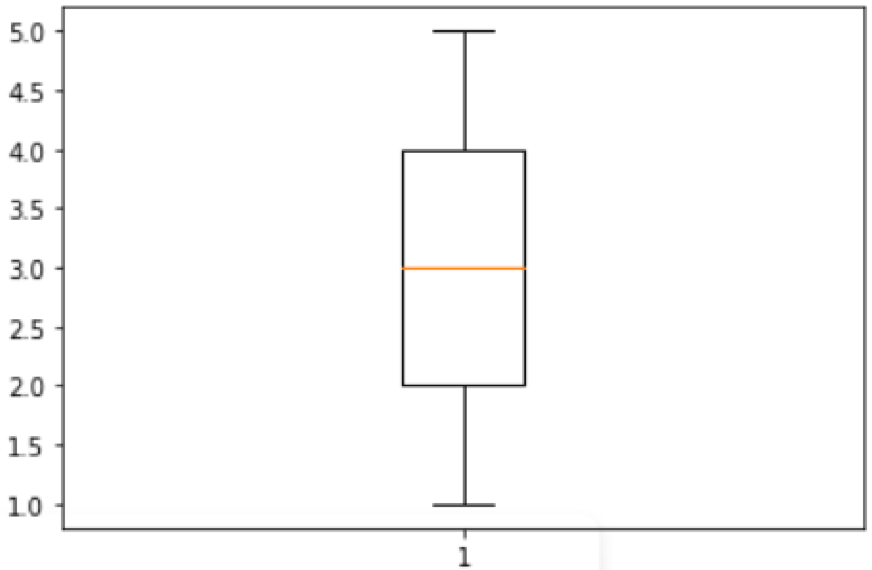
- 1 to 10 scale
- ascending order of goodness of performance
- rating is continuous

Pre processing

- Where necessary the ratings have been changed to be in an ascending scale
- **Discretization** into 5 bins for each rater using a **clustering approach**
  - sklearn function *KBinsDiscretizer* (n_bins = 5, strategy = ‘kmeans’) has been used

Classes distribution of the whole dataset after the discretization:





Pre processing

Values of attribute position ('pos_role'):

- GK (Goalkeeper) → GK (Goalkeeper)
- DR (Defender Right), DC (Defender Center), DL (Defender Left) → DF (Defender)
- DMR (Difensive Midfielder Right), DML (Difensive Midfielder Left) → DMF (Defensive Midfielder)
- MR (Midfielder Right), ML (Midfielder Left) → MF (Central MF)
- AMR (Attacking Midfielder Right), AMC (Attacking Midfielder Center), AML (Attacking Midfielder Left) → AMF (Attackig MF)
- FR (Foreward Right), FC (Foreward Center), FL (Foreward Left) → FW (Foreward)

Conversion of categorical data:
- ***One-hot encoding***

pos_AMF	pos_DF	pos_DMF	pos_FW	pos_GK	pos_MF
1	0	0	0	0	0
0	0	0	1	0	0
0	0	0	1	0	0
1	0	0	0	0	0
0	0	0	1	0	0

Pre processing
Dealing with missing values

team	pos_role	player	rater
Romania	DC	Dragos Grigore	Kicker
Romania	DC	Dragos Grigore	WhoScored
Romania	DC	Dragos Grigore	SofaScore
France	Sub	Anthony Martial	WhoScored

The player substituted another one during the game

- ➢ Creation of a new binary attribute called '**starter**'

- 'Sub' does not correspond to a role: lack of information about the player's role.
- Treated as a **missing value**

  - ➢ Filled using the mode calculated on the group of instances of the same player.
  - ➢ Some have been filled by hand
  - ➢ Some were deleted in case it was not possible to establish the role

Pre processing

Removing irrelevant attributes

- competition
- date
- match
- rater
- team

Pre processing

Removing duplicates and inconsistencies

Since there are several ratings for each performance, assigned by different newspapers, there are 2 situations:

- Agreement across different raters ⟶ duplicated instances
- Disagreement across different raters ⟶ instances with all attributes equal except class (rating)

...

win	lost	is_home_team	minutesPlayed	game_duration	starter	rating
0	1	0	90	90	1	3.0
0	1	0	90	90	1	2.0

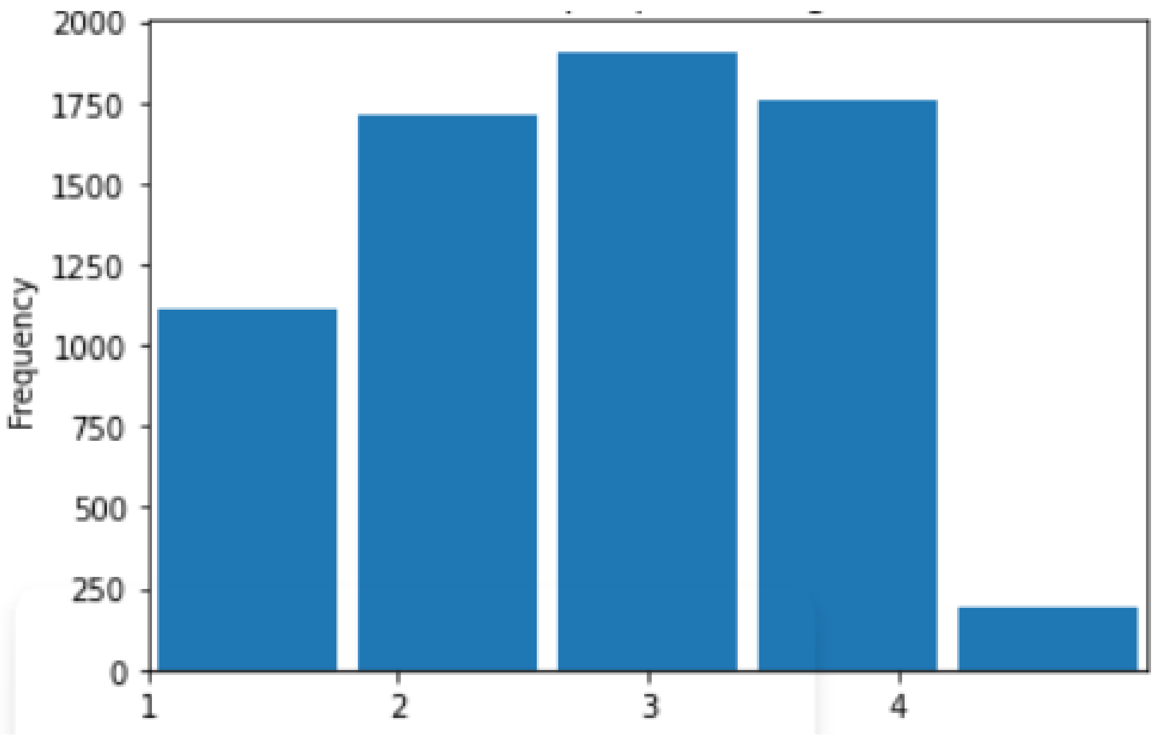
Pre processing

Normalization

Z-score normalization when algorithms that need feature scaling were tested

- function *StandardScaler* from scikit-learn

After pre processing



- 6676 instances
- 60 attributes

Classes are **imbalanced!**

Classification

Evaluation metrics:

- Focus mainly on **macro averaged mean absolute error** (provided in the *imbalanced-learn* python library), most suitable for **ordinal classification** problems where the target values are **imbalanced** [1]

All the classifiers have been evaluated with a ***10-fold cross validation***

Given the imbalanced classes, performances of classifiers were tested also after having applied over-sampling techniques for rebalancing (***SMOTE*** and ***RandomOverSampler*).**

The models have been evaluated also after **feature selection**.
Tried methods for feature selection from both scikit-learn and weka:

- *SelectKBest (*scoring function = *fclassif)*
- *SelectFromModel (*estimator = *LogisticRegression*, threshold = *median)*
- *CfsSubsetEval + BestFirst*

[1] S. Baccianella, A. Esuli, F. Sebastiani, Evaluation measures for ordinal regression, in: Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications, ISDA’09, 2009, pp. 283–287

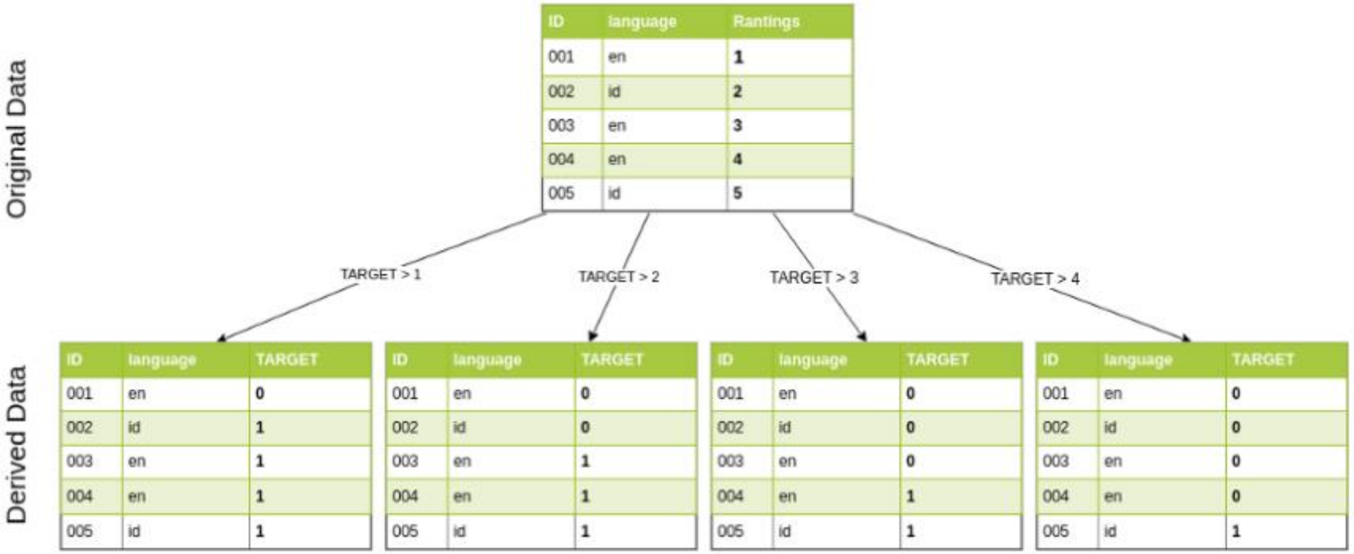
Tested models

- KNN
- Support Vector Machine
- XGBClassifier
- Random Forest
- Logistic Regression
- OrdinalClassifier (implementation of the approach proposed in [2] specifically for ordinal classification)

For each classifier was performed the hyperparameters tuning using the ***GridSearchCV*** algorithm

[2] Frank, E., Hall, M.: A simple approach to ordinal classification. Technical Report 01/05, Department of Computer Science, University of Waikato (2001)

Ordinal classification



ID	language	Rantings
001	en	1
002	id	2
003	en	3
004	en	4
005	id	5

ID	language	TARGET
001	en	0
002	id	1
003	en	1
004	en	1
005	id	1

ID	language	TARGET
001	en	0
002	id	0
003	en	1
004	en	1
005	id	1

ID	language	TARGET
001	en	0
002	id	0
003	en	0
004	en	1
005	id	1

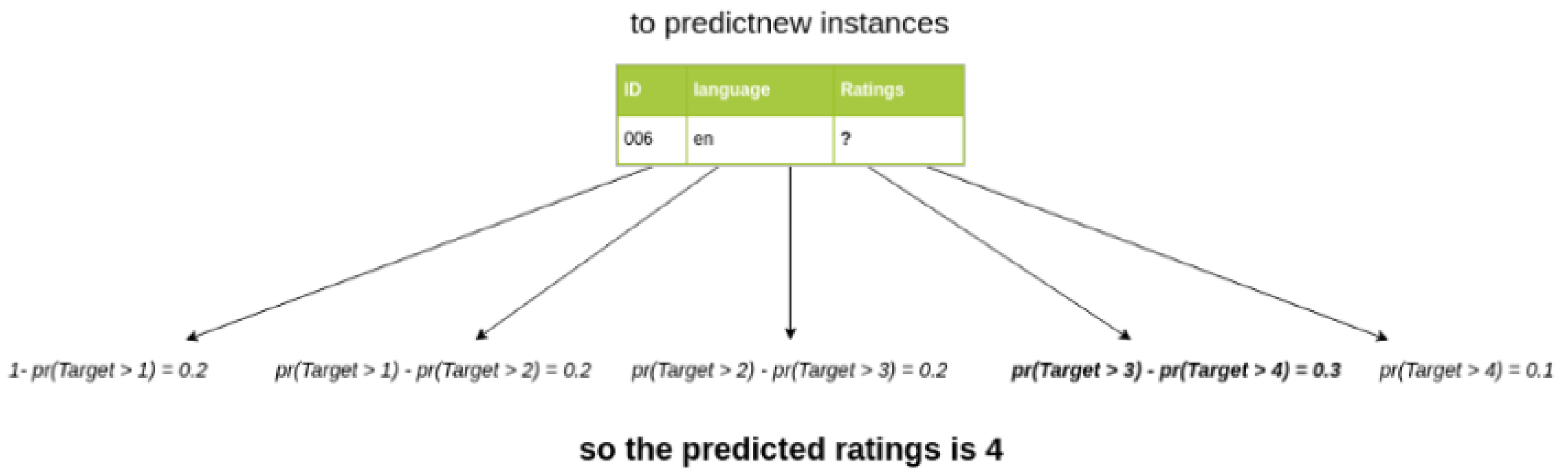
ID	language	TARGET
001	en	0
002	id	0
003	en	0
004	en	0
005	id	1

Images source: https://towardsdatascience.com/simple-trick-to-train-an-ordinal-regression-with-any-classifier-6911183d2a3c

Pr(y=1) = 1-Pr(Target > 1)
Pr(y=2) = Pr(Target>1)-P(Target > 2)
Pr(y=3) = Pr(Target>2)-P(Target > 3)
Pr(y=4) = Pr(Target>3)-P(Target > 4)
Pr(y=5) = Pr(Target>4)

The used classifiers must be able to estimate output class **probability.**

The predicted value is the one associated with the highest probability.



ID	language	Ratings
006	en	?

Results

	Attribute selection	Num. features selected	Resampling	Macro avg MAE	Avg Precision	Avg Recall	Avg F1 score	Avg Accuracy	Prediction Time
Logistic Regression	SelectFromModel	30	RandomOversampler	**0.305**	0.637	0.704	0.657	0.653	0.023
Logistic Regression	None	59	RandomOversampler	0.306	0.643	0.704	0.663	0.66	0.021
SVM	SelectFromModel	30	SMOTE	0.307	0.651	0.703	0.667	0.659	0.172
Logistic Regression	None	59	None	0.314	0.728	0.694	**0.706**	0.684	0.025
SVM	None	59	None	0.316	0.724	0.693	0.705	0.68	0.251
Logistic Regression	SelectFromModel	30	None	0.317	0.72	0.691	0.702	0.678	0.021
SVM	SelectFromModel	30	None	0.319	0.721	0.691	0.701	0.679	0.143
XGBClassifier	SelectFromModel	30	RandomOversampler	0.336	0.678	0.677	0.674	0.653	0.031
XGBClassifier	None	59	RandomOversampler	0.342	0.686	0.673	0.676	0.652	0.031
XGBClassifier	SelectKBest	28	SMOTE	0.348	0.702	0.67	0.682	0.655	0.031
XGBClassifier	CfsSubsetEval + BestFirst	**11**	None	0.348	0.611	0.671	0.633	0.601	0.02
Logistic Regression	SelectKBest	35	None	0.35	0.718	0.667	0.686	0.658	0.021
Random Forest	SelectFromModel	30	RandomOverSampler	0.353	0.699	0.662	0.674	0.65	0.05
OrdinalClassifier + DecisionTree	CfsSubsetEval + BestFirst	11	None	0.358	0.532	0.663	0.553	0.541	0.007
Random Forest	None	59	RandomOversampler	0.365	0.704	0.649	0.669	0.651	0.048
KNN	CfsSubsetEval + BestFirst	11	None	0.371	0.575	0.654	0.602	0.578	0.292

STATISTICAL SIGNIFICANCE

Student's t-test on macro-avg MAE and F1 score

α = 0.05

Chosen model for the application: *Logistic Regression* ("SelectFromModel without resampling" version)

	Macro avg MAE		F1 score	
Logistic regression w/o attribute selection – **Logistic regression** Attr. Selection = *SelectfromModel (30 attr. selected)*	**0.314** 0.317	*p* = 0.718	0.706 **0.702**	*p* = 0.388
Logistic regression *SelectfromModel + Resampling* – **Logistic regression** Attr. Selection = *SelectfromModel (30 attr. selected)*	**0.305** 0.317	*p* = 0.065	0.657 **0.702**	*p* = 0.001
XGBClassifier Attr. Selection = *CfssubsetEval + BestFirst (11 attr. Selected)* – **Logistic regression** Attr. Selection = *SelectfromModel (30 attr. selected)*	0.348 **0.317**	*p* = 0.02	0.633 **0.702**	*p* = 1.23e-05

The application

Functional requirements

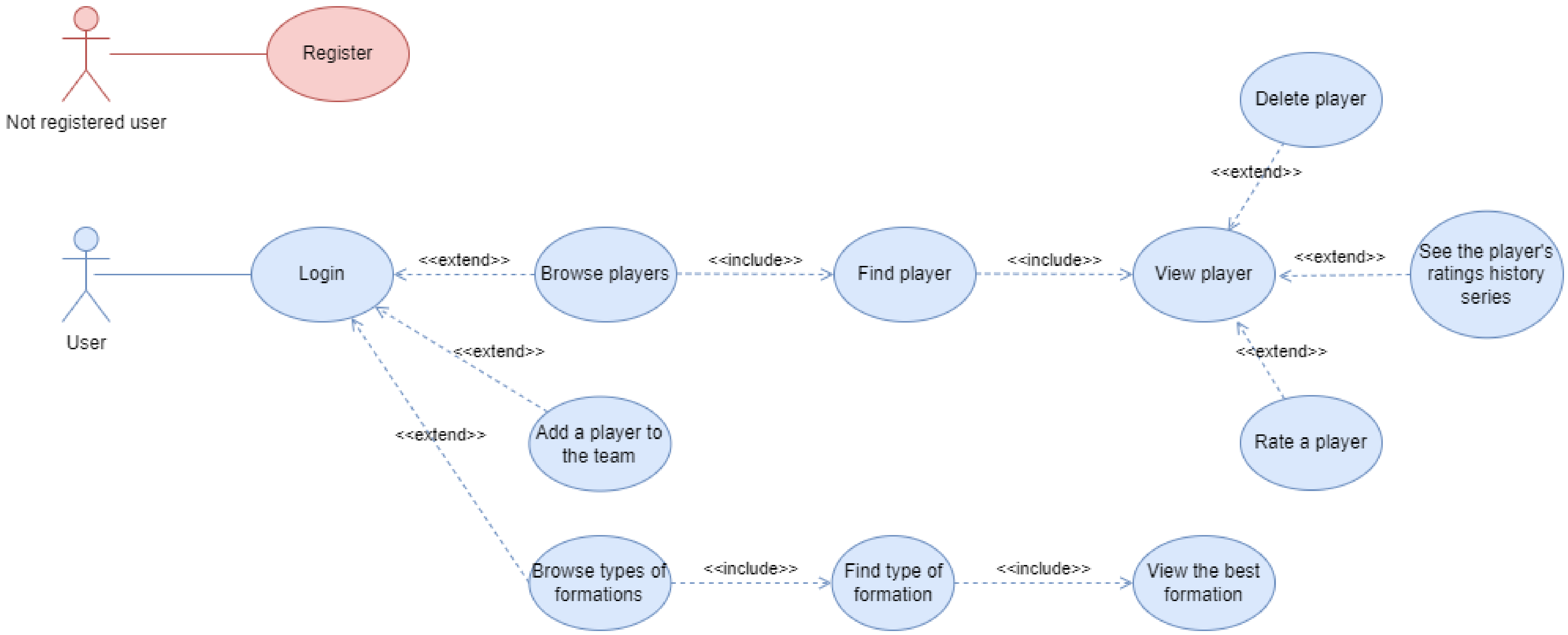
A User can:

- Login
- Create his own team and modify it, adding new player or removing others
- Rate a player entering statistical information about a player's performance in a match
- See the best formation for the next match suggested by the application
- See the player’s ratings history series through a graph.

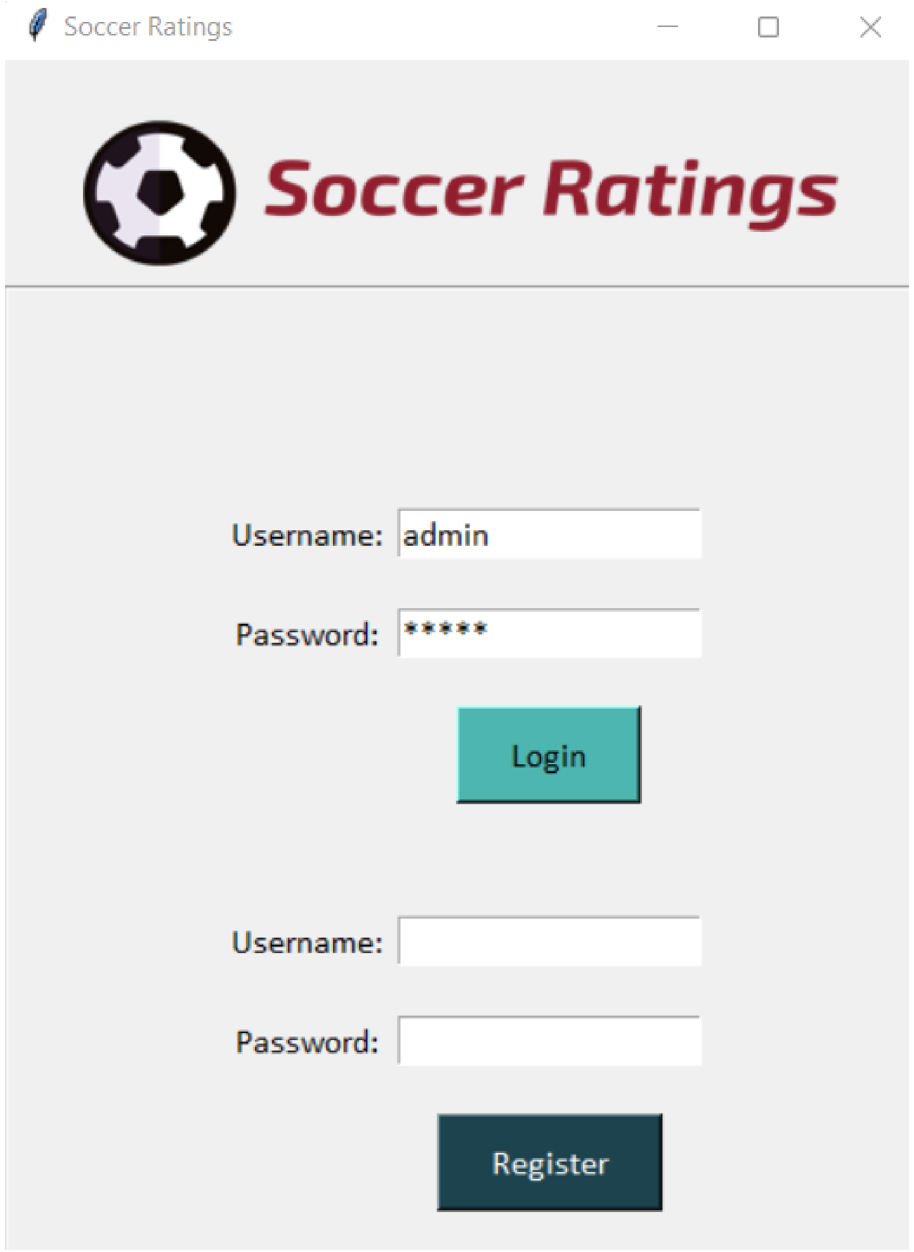
Non-functional requirements

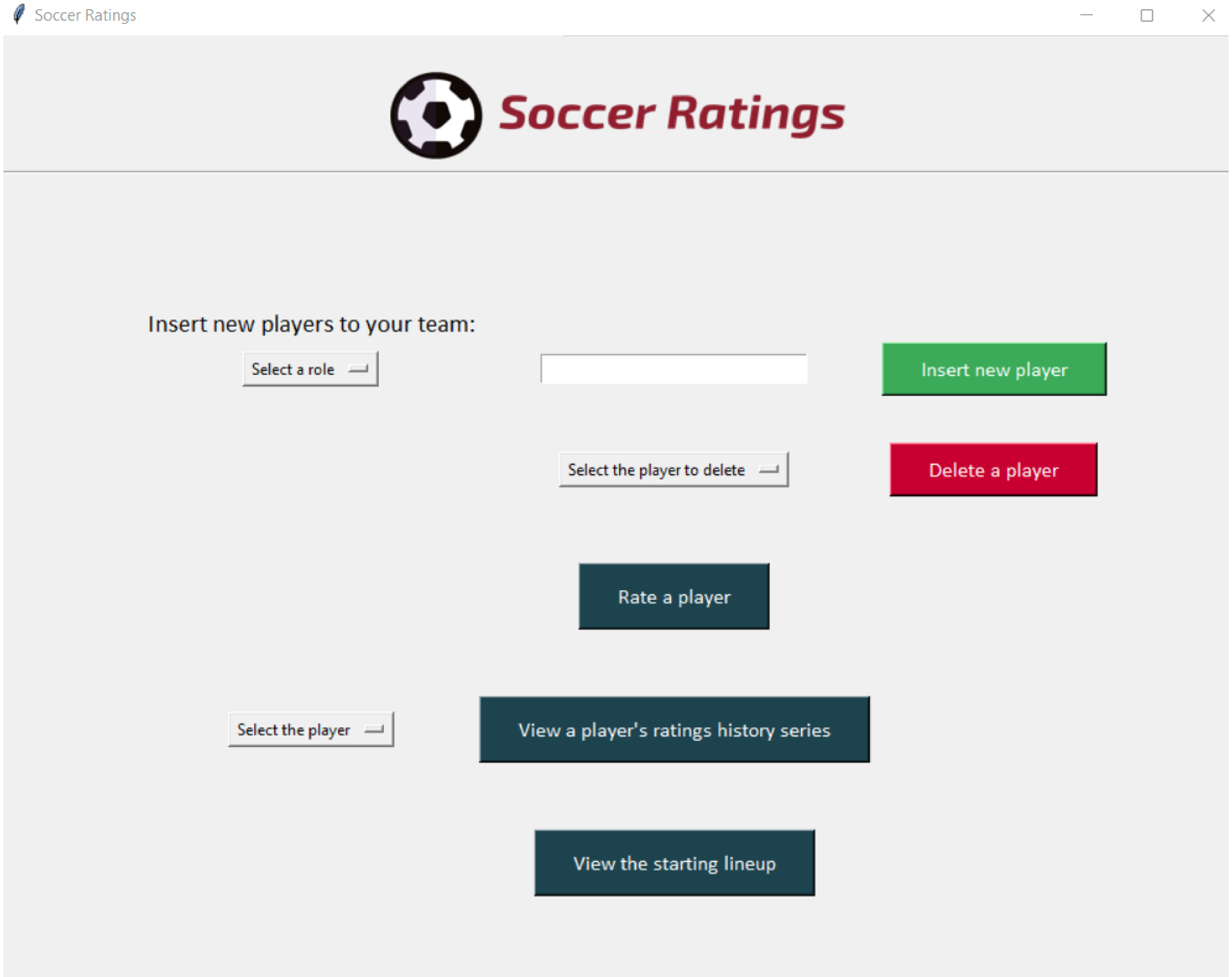
- The application must be user-friendly and easy to use through a clear user interface.
- The application must provide fast responses to users requests

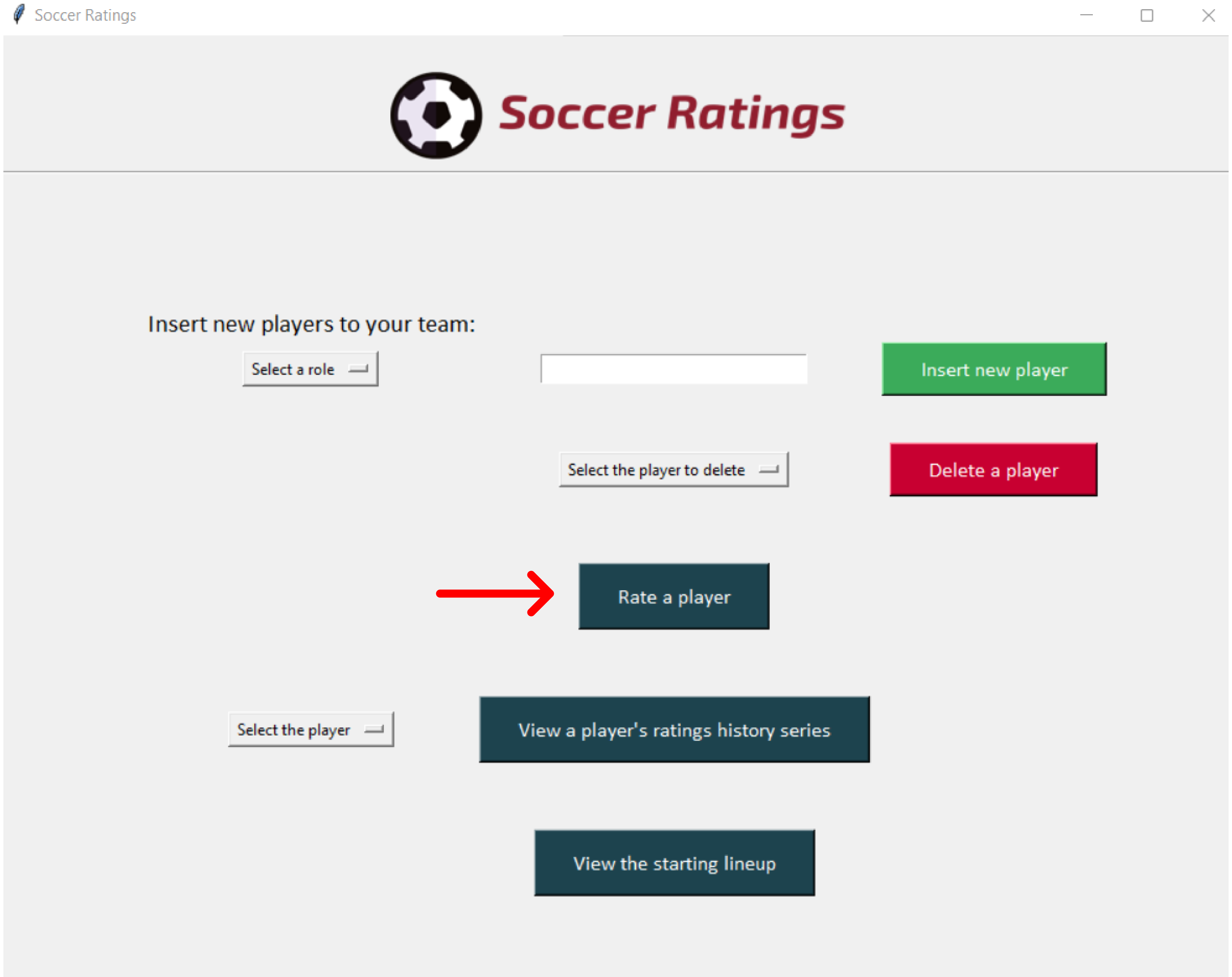
UML Use case diagram

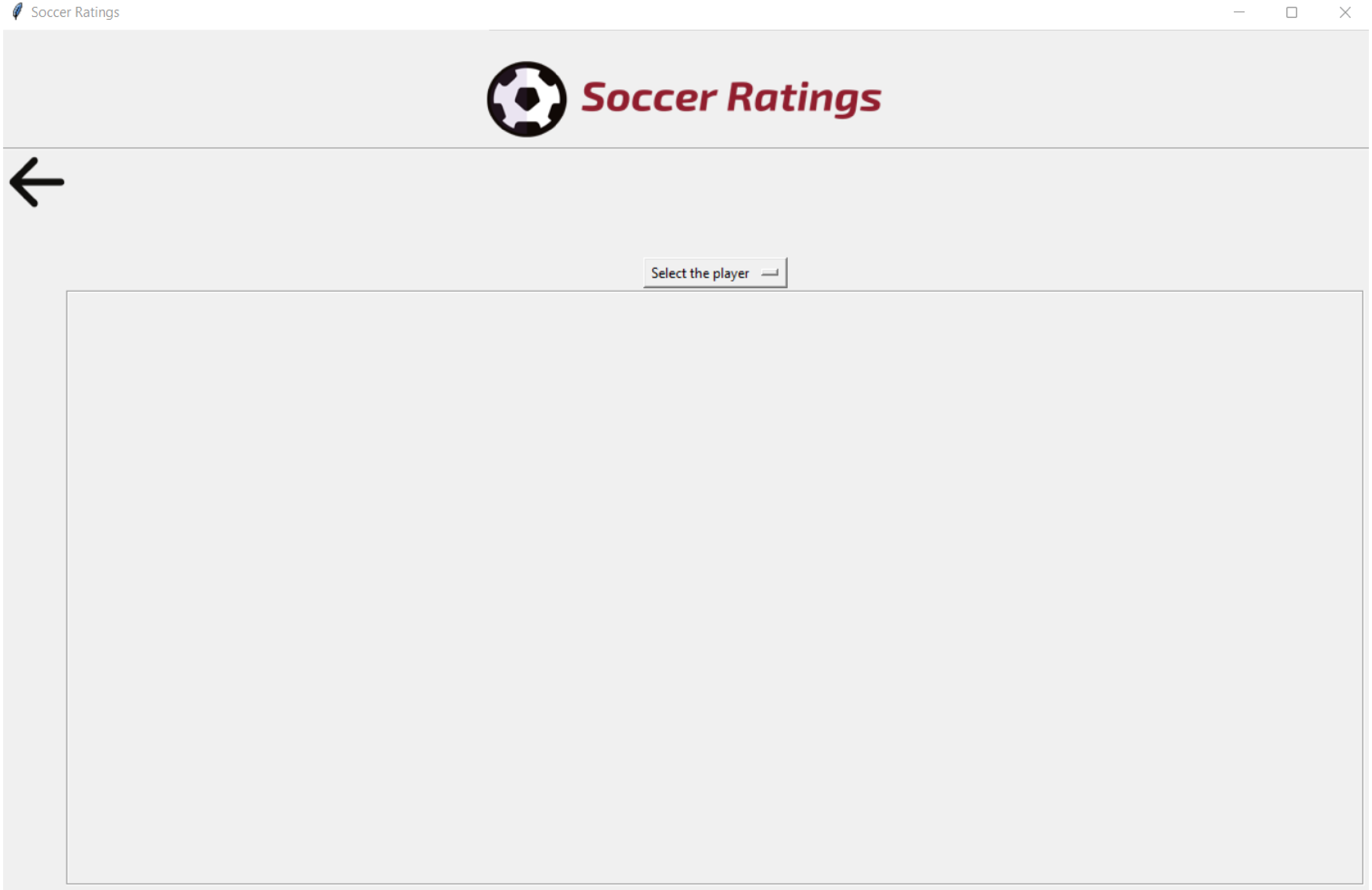


Login/registration




Soccer Ratings
Soccer Ratings
Insert new players to your team:
Select a role
Insert new player
Select the player to delete
Delete a player
Rate a player
Select the player
View a player's ratings history series
View the starting lineup


Soccer Ratings
Soccer Ratings
Insert new players to your team:
Select a role
Insert new player
Select the player to delete
Delete a player
Rate a player
Select the player
View a player's ratings history series
View the starting lineup


Soccer Ratings
Soccer Ratings
Select the player

Soccer Ratings

Soccer Ratings

Marco Verratti

Select the date of the match

4/11/22

Goals

Assists

Key passes

Dribblings

Touches

Accurated passes

Accurated crosses

Accurated long passes

Stopped shots

Ground duels won

Aerials won

Possession lost

Clearances

Interceptions

Tackles

Shots on target

Own goals

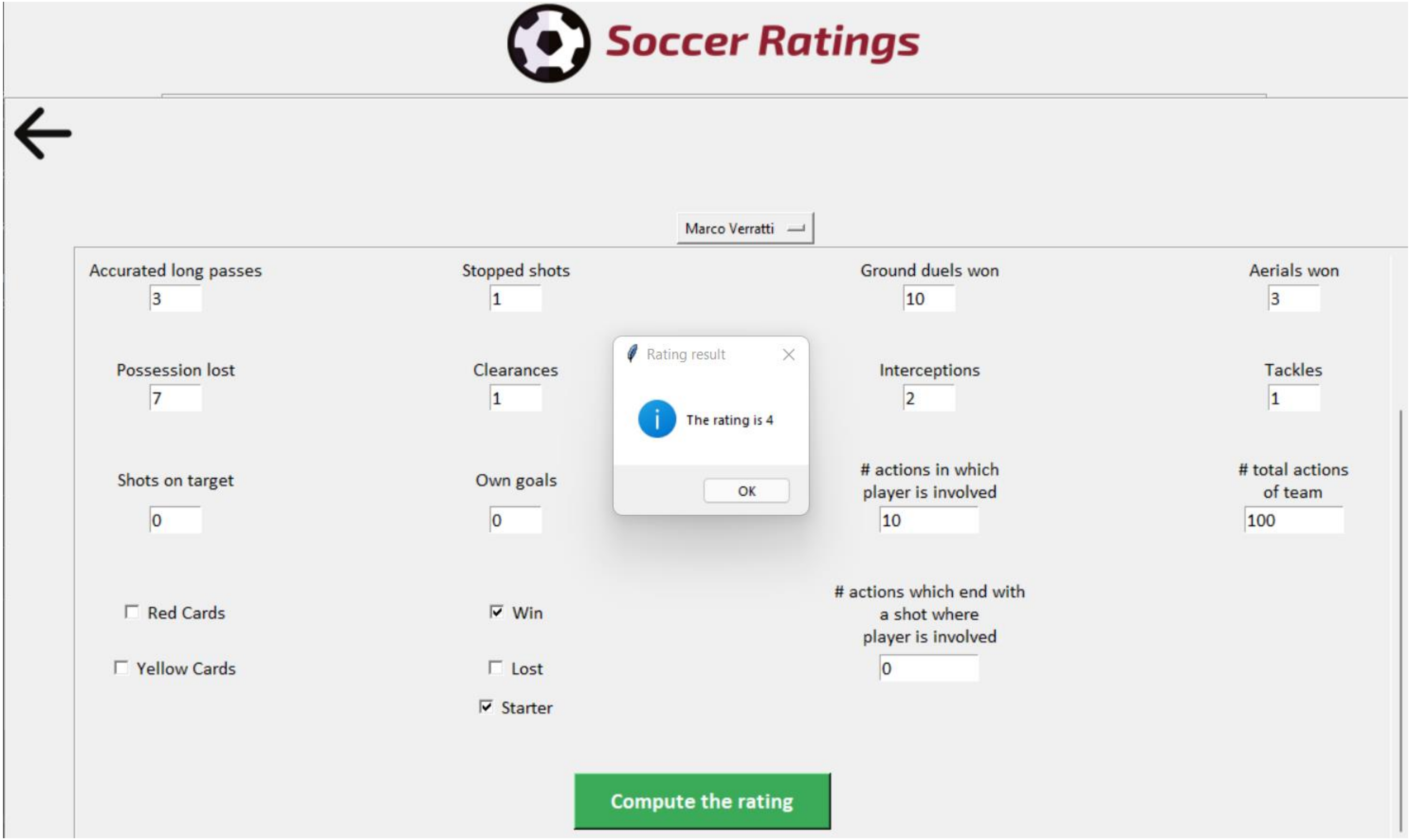
actions in which player is involved

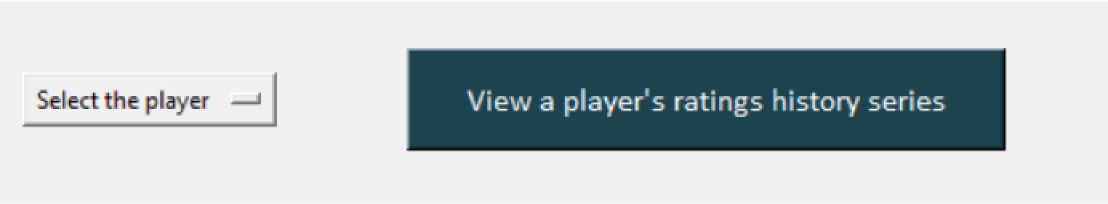
total actions of team

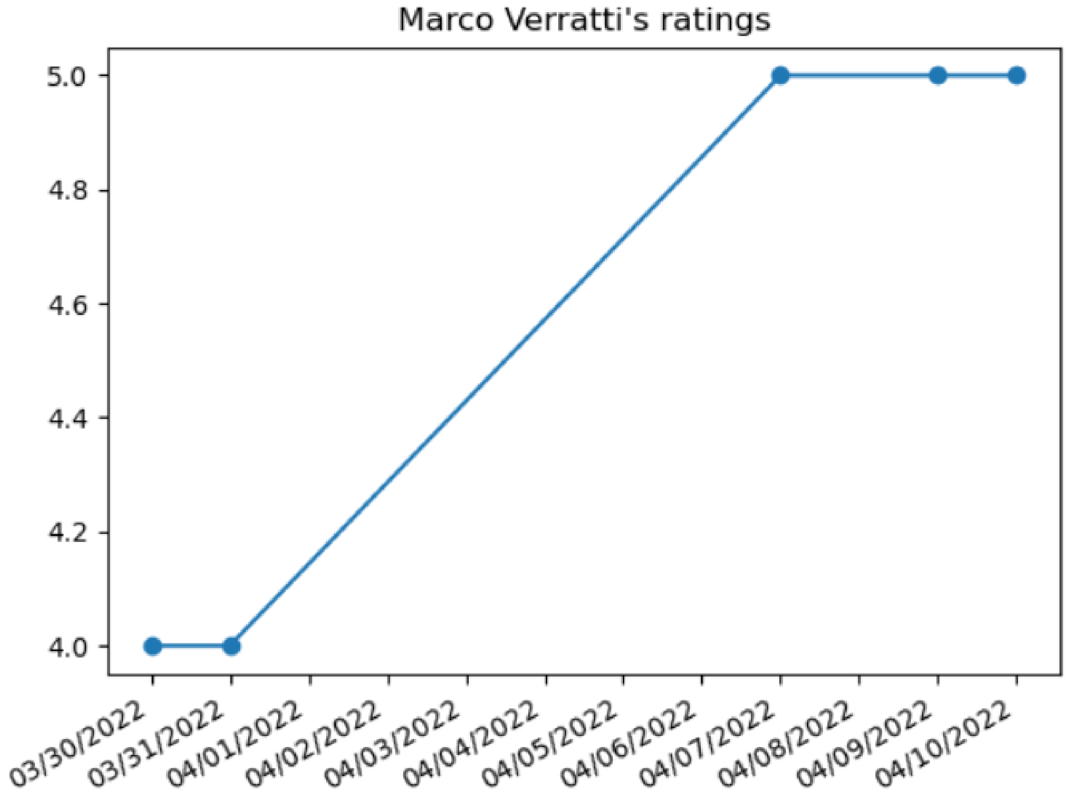
☐ Red Cards

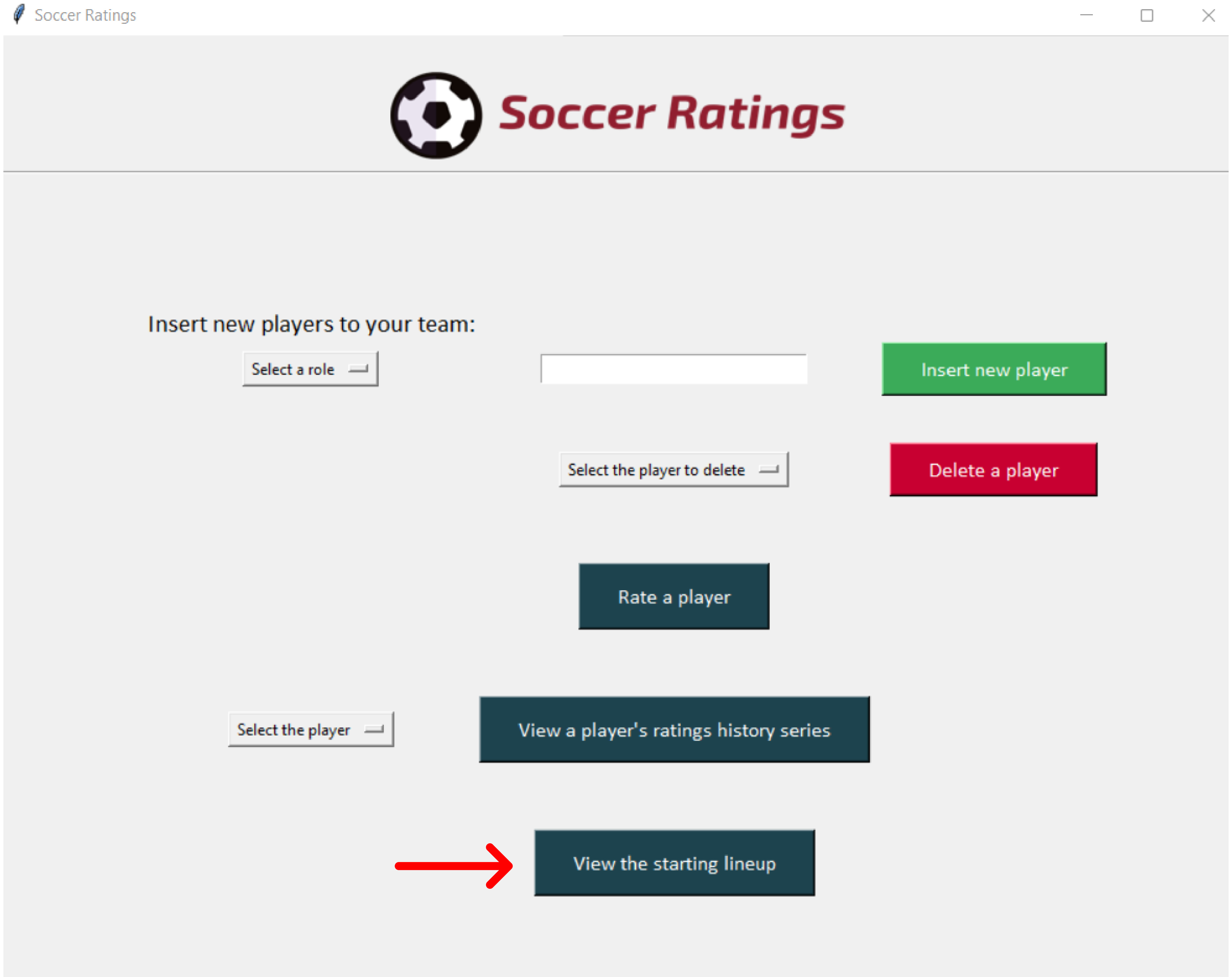
☐ Win

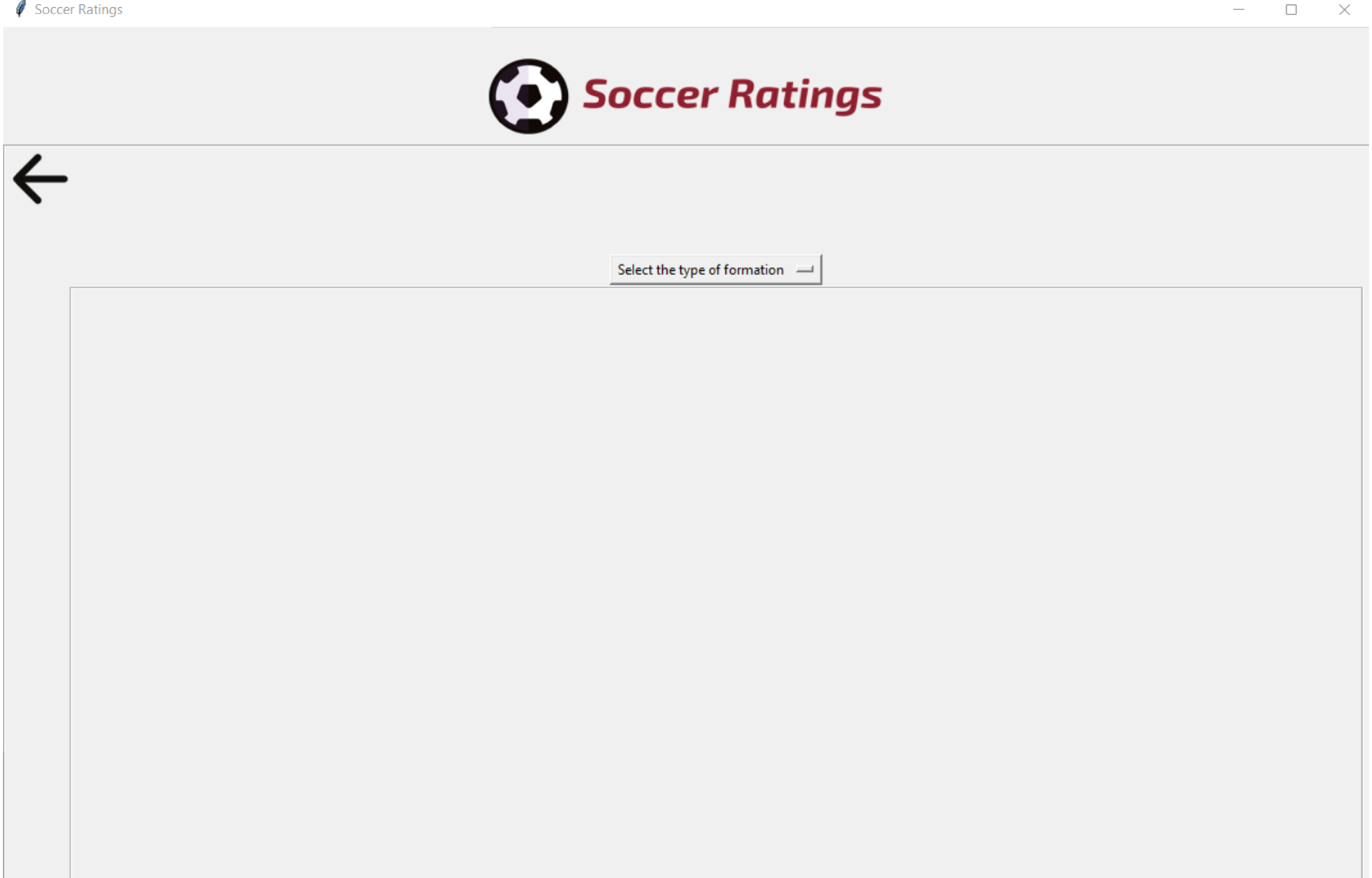
actions which end with a shot where player is involved


Soccer Ratings
Marco Verratti
Accurated long passes
3
Stopped shots
1
Ground duels won
10
Aerials won
3
Possession lost
7
Clearances
1
Interceptions
2
Tackles
1
Shots on target
0
Own goals
0
actions in which player is involved
10
total actions of team
100
Red Cards
Win
Yellow Cards
Lost
Starter
actions which end with a shot where player is involved
0
Compute the rating
Rating result
The rating is 4
OK


Select the player
View a player's ratings history series


Marco Verratti's ratings
5.0
4.8
4.6
4.4
4.2
4.0
03/30/2022
03/31/2022
04/01/2022
04/02/2022
04/03/2022
04/04/2022
04/05/2022
04/06/2022
04/07/2022
04/08/2022
04/09/2022
04/10/2022


Soccer Ratings
Soccer Ratings
Insert new players to your team:
Select a role
Insert new player
Select the player to delete
Delete a player
Rate a player
Select the player
View a player's ratings history series
View the starting lineup


Soccer Ratings
Soccer Ratings
Select the type of formation

Soccer Ratings

Soccer Ratings

4-4-2