جامعة حلوان
HELWAN UNIVERSITY
Faculty of Engineering
Computer Engineering Department

Player Performance Predictor

Project Team Members:

Omar Sayed Ahmed
Abdulrahman Mahmoud Abdel-Aal
Abdul Aziz Shaaban Abdul Aziz
Amr Aly Abdulrahman
Ali Abdelwahab Salah

Under Supervision of:

Professor Dr. Manal Shoman

July 2021

ACKNOWLEDGMENT

We would like to express our sincere gratitude to several individuals and organizations for supporting us throughout our Graduate study.

First, we wish to express our sincere gratitude to our supervisor, Professor Dr. Manal Shoman, for her enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped us tremendously at all times in our research and writing of this thesis. His immense knowledge, profound experience and professional expertise in Data Quality Control has enabled us to complete this research successfully. Without her support and guidance, this project would not have been possible. We could not have imagined having a better supervisor in our study.

We also grateful to all the Teaching staff in the Computer Engineering Dep., for their consistent support and assistance. It was great sharing premises with all of you during the last five years.

Thanks for all your encouragement!

Table of Contents

Acknowledgment 2
Table of Contents 3
Table of Figures 6
Introduction: 8
Abstract: 10
Motivation 11
Problem Statement 11
Project Phases 12
Identify the problem 12
Research the problem 13
Formulate a hypothesis 13
Conduct an experiment 14
Reach a conclusion 14
Project Architectural 15
Literature Review: 18
1- Assessment and Evaluation of Football Performance - Feb 2000 18
2- Sports Analytics for Football League Table and Player Performance Prediction 23
3- Article Evaluation of the Technical Performance of Football Players in the UEFA Champions League 30
Analysis and Requirements 35
1. Requirements: 35
A. Functional Requirements 35
B. Non Functional Requirements 36
2. FUNCTIONAL Requirement Specification: 36
A. Stakeholders 36
B. Actors and Goals 37
C. Use Case 38
I. Use Case Description 38
ii. Use Case Diagram: 39

D. System Sequence Diagram 40
All users search for player data 40
Coach and scout get performance prediction 41
Coach gets player rating 42
E. Class Diagram 43
Implementation 44
Dataset and Features 44
Data Analysis 44
Gathering Data 44
Cleaning Data 47
Assess Data 47
Direct assessment 49
Indirect Assessment 50
Assess the Quality of Data 51
Rating Model 52
Prediction Model 55
Mobile Application 60
Technical Specification 60
Overview 60
Installation 61
Packages 62
Getting started Database 63
Rapid API-Football 63
Application User Journey 69
Home page 69
Leagues page: 70
Team's page: 71
Squad page: 74
Stats page: 75
Search page: 78
Video Analysis 81
Video Analysis Model 84
Tools and Technologies 90
Programming Languages: 90
Online Tools 90

Libraries 91
Data Analysis 91
Data Visualization 91
Data Splitting 91
Deep Learning Framework 92
Arrays and matrices calculation 92
Database 92
Models 92
Data API 92
CONCLUSION, Results and Future work 93
Conclusion 93
Results 94
Future Work 94
References 97

Table of Figures

Figure 1-Project Phases 12
Figure 2- Project Structure 15
Figure 3- Quality rating 20
Figure 4- Flexibility test 21
Figure 5-Process 23
Figure 6- Regression results 27
Figure 7- Linearity of model 27
Figure 8- Features of positions 32
Figure 9- Features of 3 positions 32
Figure 10- Use case 39
Figure 11- sequence diagram1 40

Figure 12- sequence diagram2 41
Figure 13- sequence diagram3 42
Figure 14- class diagram 43
Figure 15- events 45
Figure 16-events types 46
Figure 17- parameters 46
Figure 18- final data 47
Figure 19- Rating model 1 52
Figure 20- Linear regression 53
Figure 21-Rating model 2 53
Figure 22- Rating model 3 53
Figure 23- Rating model 4 53
Figure 24- actual vs predicted 54
Figure 25- import 55
Figure 26- Read Player Data 55
Figure 27- features optimization 56
Figure 28- final features 56
Figure 29- Rating 56
Figure 30- Train 57
Figure 31- Scaling 57
Figure 32- Predict 57
Figure 33- Actual Vs Predicted 58
Figure 34- Accuracy 58
Figure 35- Prediction Model 59
Figure 36- Prediction for ID 59
Figure 37- App packages 62
Figure 38- APP with API 63
Figure 39- connection to API 64
Figure 40- connection to API 65
Figure 41- connection to API 66
Figure 42- Connection to API 67
Figure 43- Connection to API 68
Figure 44- Home Page 69
Figure 45- League Page 70
Figure 46-Teams 71
Figure 47- Teams 72
Figure 48- Teams 73
Figure 49- Squad 74
Figure 50-Stats 75
Figure 51- Predict 76
Figure 52- Stats 77
Figure 53- Search 78
Figure 54-Search 79
Figure 55-Search 80
Figure 56- Video detection 82
Figure 57-Video detection 82
Figure 58- Video detection 83
Figure 59- Libraries Import 85

Figure 60- Classification model 85
Figure 61- Yolo Model 85
Figure 62- Object detection 86
Figure 63-Draw 87
Figure 64-Implementation 88
Figure 65- Implementation - 2 89
Figure 66- Actual vs predicted 94

INTRODUCTION:

Common Machine Learning applications in sports analytics relate to player injury prediction and prevention, potential skill or market value evaluation, as well as team or player performance prediction. This report focuses on football. Its scope is player performance prediction. A reliable prediction of the player performance score prediction for certain teams & leagues is

presented, using past data and advanced statistics. We approach detection and recording of personal skills and statistical categories that separate an excellent player from an average football player in a different squad position. Experimental results range between encouraging to remarkable, especially given that predictions were based on data available at the beginning of the season

Long–term performance prediction for teams or individual players are fields requiring exploration. Not only coaches, but also sports agents and bookmakers are interested in how players perform during a season compared to previous ones. What is discussed in this section is the context of this problem. Also, the objectives of the research are set. The unique components of football matches make long–term predictions very difficult; only few goals are scored per match. Also, there is no clear changeover between the instantaneous change of possession and transition between offense and defense. Moreover, player positions and tactics are not fixed and finally, the game has a continuous flow, which complicates recording of game events

Our project focuses on statistics from the previous seasons and historical data. The novelty of this project is that advanced metrics were used to predict next season performance before the season begins, not after matches have already been played and recorded. Additionally, attackers are usually graded higher than defenders, even if they are not always more

influential in team strategy. So, regarding player evaluation, this research attempts to identify skills and features suitable that make good defenders.

Abstract:

Sports analytics is the use of historical data and advanced statistics to measure performance, make decisions and predictions regarding performance and outcomes, in order to gain an advantage over competitors. Performance prediction is the commonest task in sports analytics. Sport analyst's process data regarding players for the prediction of team and individual player efficiency. Forecasts may be related to

short–term or long–term events. For that reason, diverse methods and algorithms have been deployed. Clubs use sophisticated devices and software (i.e. GPS tracking systems) to gather and analyze data generated by players during training sessions and matches. They process these data to use for short–term decision making and long–term organization development. Also, extensive analysis of all data available is a prerequisite for betting companies. Fans are also very interested in advanced statistics and how they affect football.

For all the above reasons, the use of sports analytics has increased during the last few years. Football was selected for our research because of the abundance of statistical categories and historical data, its fame, as well as the simplicity of its rules and of national championships formats. On the other hand, there are special difficulties, which make football long–term prediction challenging.

The nature of football makes statistical recording of match events as well as player and team rating, an ambiguous process.

Motivation

Football being one of the world's most popular games has a craze in everyone's mind. A football Fan must have wanted to know the results before the game at least once in their lifetime! In sport Prediction, large numbers of features can be collected including the historical performance of the teams, results of matches, and data on players, to help different

stakeholders understand the odds of Winning or losing forthcoming matches. The aim of this research is predicting football matches using deep learning algorithms such as DNN and RNN.

Problem Statement

Competitive sport is a number game, and modern athletic teams are truly beginning to embrace this fact. We are seeing a surge of data driven jobs in the back office of myriad different franchises, and people are beginning to wonder just how effective cutting edge AI, ML and DL techniques can be when applied to athletic contexts. Sports have always been a purely human endeavor and there is an unarguable element of randomness and chance that dictates who the victor will be on any given day but this begs on an interesting question: is there an underlying pattern to this randomness?
In this project we focused especially on matches in Premier League (England Football league). Few reasons made us work on that topic. We are football fans and also passionate about the area of statistical modelling and prediction. Working on this topic was a good opportunity to join these two interest areas.
Here, what exactly did we predict? For each game, we will predict 3 probabilities: probability that the home team wins, probability that the result is a draw and probability that the home team loses the match. These predicted probabilities could be useful in evaluating our models' accuracies by forecasting the result of each match, which will be associated with the biggest predicted probability.

We used logistic regression, random forest classifier, a 2-layer neural network and an LSTM to output a predicted result which could take any of the three options: a home win, a draw, or win, or an away win

Project Phases

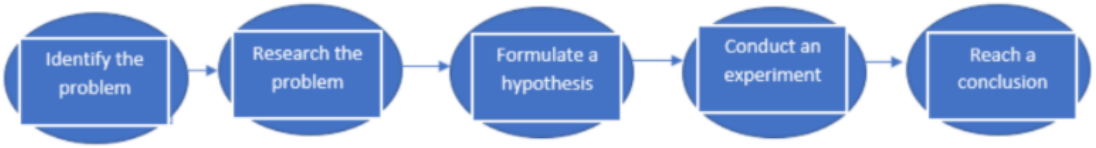


Figure 1-Project Phases

Identify the problem

Stating the problem and the topics we will work on by defining the major parts related to the problem we are facing. Deciding what is our goal from the project? And for our case the goal is to obtain a high accuracy model to rate and predict the performance of football players. In another way, how you will design a model which will rate each player depend on the position features of the player and to design a model which will predict the player performance in the future depend on the past seasons rates.

Research the problem

Collecting data and information about similar practice and articles to state the problem and from Text-books, scientific papers and other useful resources and Searching about the different techniques, methods and Approaches used to solve the problem or related ones. In our

Case we read books about computer vision and machine learning
And its models and techniques and we reviewed a lot of papers
Concerned with Human Action Recognition and Violence
Detection and we dived into the different datasets regarding
HAR to find the appropriate ones to use.

Formulate a hypothesis

Defining a theoretical hypothesis to solve our problem. This done by designing a two ML models for rating and prediction and both of them contains the required features to measure the performance of the players in each position in the game and this depend on a previous researches and studies which realized that those features are the best to rate.

Conduct an experiment

Testing the hypothesis by conducting an experiment to find out whether this hypothesis is working or not. In our case this was done by applying the designed models on previous data which its rates are exist to compare the results and be sure that the model features are the optimum ones.

Reach a conclusion

Determining the conclusion based on the obtained result from the conducted experiment for the assumed hypothesis. For our project we

evaluated the result by measuring the different accuracies obtained from the different machine and deep learning models and comparing them by other accuracies obtained from scientific papers that use different data Representations and other models on the same datasets we used. And we found a significant increase in the accuracies from our models.

Project Architectural

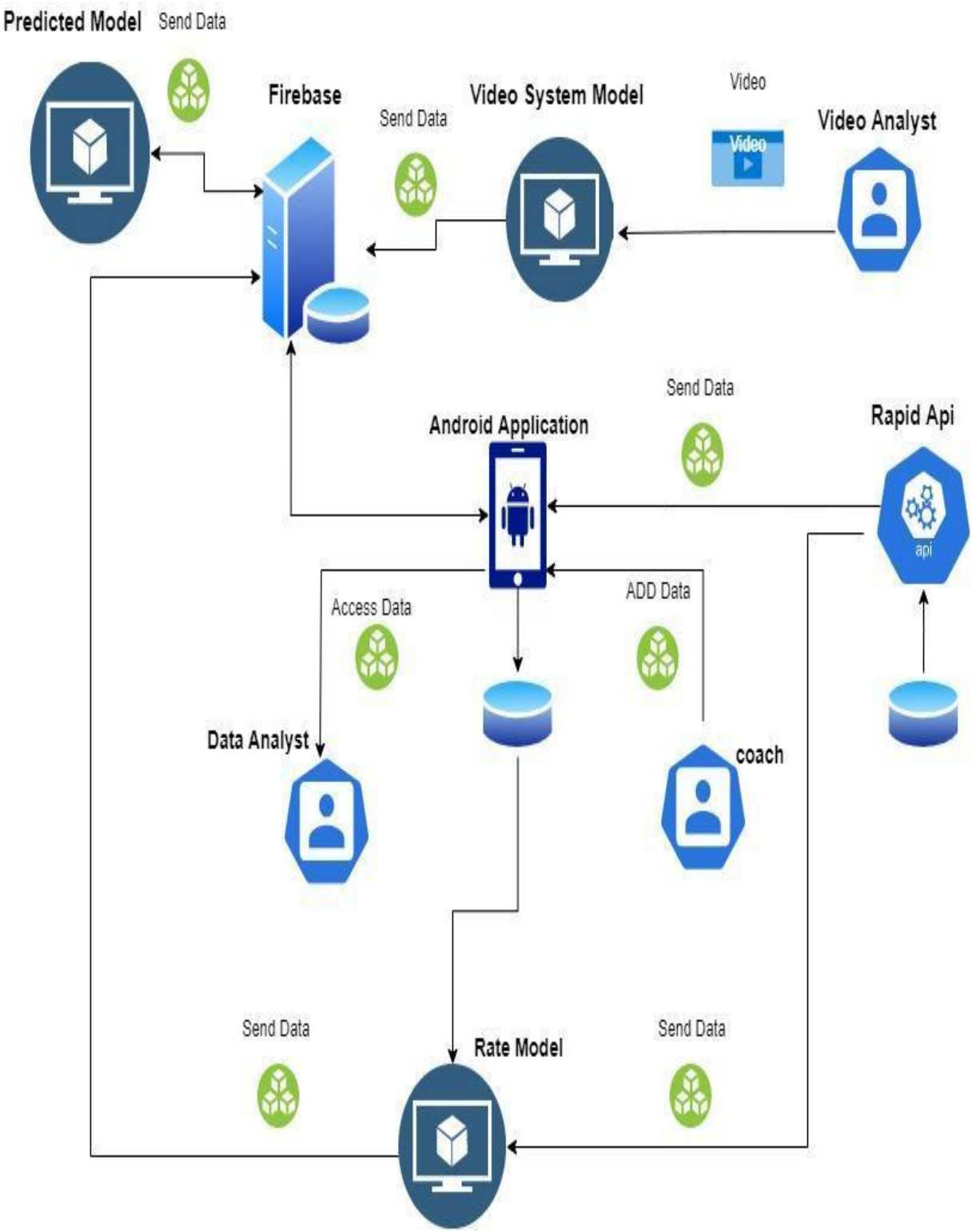


Figure 2- Project Structure

In the system architecture there's a several methods of getting players data and statistics:

The first way is the video analysis: through a real time sports AI data collection Camera, player's movements, skills, behavior and data are being recorded as an event then stored and passed to the video system model.

In the video system model all those events translated to a certain player statistics which was already predefined to this model earlier.

A rating and prediction models are included in this analysis model which will do the required calculation for player rating and prediction process then store those data in the database which in our system here (google firebase)

The second way will be through the API integration: a player's statistics data stored in one of the most famous and trusted API providers will be collected through the API integration and in this case these data will go through two paths:

The first path is to be visualized on the application in case the users requested to just monitor and visualize a certain Numeric statistics for the player.

The second path is that those data will go directly to the rating model which we designed according to the most effective parameters on the player rating process.

After we rate each player depending on his position and role parameters we stored all of those rates on the database, so we can present those rates through the application.

In this stage the prediction model can use all the rates of players in different seasons which we store earlier in the database to predict the next season rates.

Those predicted future rates can be now displayed through the application.

The third way of data input to the system is through the coaches and players analysts by using the functionality of player statistics entry.

And those data received from the user will be saved into the application database then go through the same process of player performance rating and prediction.
At the end of all of these different processes, all the different users with different roles can visualize and predict and have the analysis for the players.

Literature Review:

1- Assessment and Evaluation of Football Performance - Feb 2000

By: Dieter Ro¨sch, MA, Roy Hodgson,Lars Peterson,MD, Toni Graf-Baumann, MD, Astrid Junge,i PhD, Jiri Chomiak,a MD, and Jiri Dvorak,i b MD
From : the Department of Sport and Sport-Pedagogy, Pedagogic-University, Freiburg, Germany, the Grasshoppers Football Club, Zurich, Switzerland, Gothenburg Medical Center, Vastra Fro¨lunda, Sweden, the Office for Management in Medical Research, Tenningen, Germany, the Schulthess Clinic, Zurich, Switzerland, and the aOrthopedic Clinic Bulovka, Praha, the Czech Republic

This Research considered **the most important variables for measuring performance in team sports such as football are physical condition and technical and tactical performance**. However, because of the complexity of the game of football it is difficult to ascertain the relative importance of each of these variables. **The aim of the present study was to develop a standardized test battery to evaluate physical performance in football players**.

The F-MARC test battery was designed to closely relate to the football player's normal activity and comprised a functional, structured training session of approximately 2.5 hours.
It included a "quality rating" of the warm-up procedure, tests of flexibility, football skills, power, speed, and endurance. The players finished with a cool-down. **A total of 588 football players underwent the F-MARC test battery**. Mean values for performance on each test are presented for groups of differing age and skill levels. The test battery proved to be a feasible instrument to assess both physical performance and football skills.

This study supports the proposal by Balsom (1994) that analysis of an individual player's physical profile, in relation to mean values for a similar age group and skill level, might be of assistance to the coach in objectively evaluating the effects of a specific training program. It may also be of use to the physician and physical therapist responsible for monitoring progress during rehabilitation after football injuries.

In this research **Players were grouped according to the age and skill level of their team**. Adult players were categorized into four skill level groups: top-level adults (first and second league), third league, amateur teams (Division), and local teams.
Youth players were divided into two age groups (14- to 16-year-old players and 16- to 18-year-old players) and two skill-level groups (high and low).

The features which was the measurement of the performance of players in this research are:
Warm-up Rating, Flexibility Tests, Football Skills, Power, Speed, and Endurance Tests

All data were processed on a Macintosh computer (Apple Computers, Cupertino, California) using Microsoft Office (Microsoft Corp., Redmond, Washington). The statistical procedures were accomplished using Stat View (version 4.5; SAS Institute Inc., Cary, North Carolina) and SPSS

(Version 6.1; SPSS Inc., Chicago, Illinois). Methods applied were frequencies, cross-tabulations, descriptive, and means. Differences between groups were examined by t-tests. Unless otherwise stated, only results that were significant at the 5% level or less are presented

Samples:

A total of 588 players from Germany, France, and the Czech Republic were examined. The ages of the players ranged from 14 to 41 years (mean, 18.4; SD, 4.0)

TABLE 1
Quality Rating for Warm-up in Football Players of Different Age and Skill Levels

Rating	Adult players								Youth (16–18 years)				Youth (14–16 years)			
	Top-level (N = 40)		Third league (N = 41)		Amateur teams (N = 25)		Local teams (N = 38)		High-level (N = 137)		Low-level (N = 96)		High-level (N = 118)		Low-level (N = 93)	
	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)
Cardiopulmonary (1 = good to 5 = poor)	2.7	(1.2)	2.9	(1.3)	3.8	(0.9)[c]	3.0	(1.2)	1.8	(0.9)	2.1	(0.9)[a]	2.2	(1.0)	2.1	(0.9)
Muscular (1 = good to 5 = poor)	1.8	(1.0)	2.1	(1.0)	3.3	(1.0)[c]	2.9	(1.0)[c]	1.9	(0.9)	2.8	(0.8)[c]	2.3	(1.0)	2.7	(1.0)[b]
Coordination (1 = good to 5 = poor)	1.7	(0.7)	1.9	(0.9)	2.7	(0.7)[c]	2.4	(0.8)[c]	1.7	(0.8)	2.5	(0.7)[c]	2.2	(0.5)	2.5	(0.7)[b]

[a] Significance of difference from top or high-level players of the same age group: $P \leq 0.05$.
[b] Significance of difference from top or high-level players of the same age group: $P \leq 0.01$.
[c] Significance of difference from top or high-level players of the same age group: $P \leq 0.001$.

Figure 3- Quality rating

TABLE 2
Tests of Flexibility in Football Players of Different Age and Skill Levels

Test	Adult players								Youth (16–18 years)				Youth (14–16 years)			
	Top-level (N = 40)		Third league (N = 41)		Amateur teams (N = 25)		Local teams (N = 38)		High-level (N = 137)		Low-level (N = 96)		High-level (N = 118)		Low-level (N = 93)	
	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)	Mean	(SD)
Active raising right leg in a supine position (deg)	77.9	(8.7)	77.2	(9.9)	71.4	(8.6)[b]	74.3	(7.1)	73.2	(8.4)	74.7	(9.0)	73.5	(9.5)[a]	70.3	(8.7)
Passive raising right leg in a supine position (deg)	90.7	(9.5)	95.5	(9.4)[a]	86.9	(8.9)	92.5	(12.3)	88.9	(10.6)	88.2	(11.2)	88.0	(12.5)	86.8	(11.4)
Active raising left leg in a supine position (deg)	75.9	(7.7)	78.6	(7.1)	72.4	(13.2)	79.4	(8.9)	75.5	(8.0)	75.1	(8.6)	75.1	(10.0)	74.0	(9.6)
Passive raising left leg in a supine position (deg)	88.3	(7.6)	90.0	(5.7)	82.3	(12.6)[a]	90.5	(10.9)	89.3	(9.4)	85.8	(9.7)[b]	87.5	(10.4)	84.9	(10.4)
Bending to the right side (cm)	25.1	(3.3)	25.5	(4.2)	25.6	(3.7)	26.0	(3.7)	26.8	(3.5)	26.1	(3.6)	26.3	(4.2)	25.4	(4.1)
Bending to the left side (cm)	25.1	(3.5)	25.4	(3.7)	24.9	(3.9)	25.0	(3.1)	26.5	(3.4)	25.9	(3.7)	25.7	(4.4)	24.9	(4.1)
Leg splits forward, right leg to ground (cm)	38.2	(3.0)	38.0	(3.6)	39.6	(2.8)	38.4	(3.8)	38.3	(3.6)	39.1	(3.4)	38.5	(3.3)	38.5	(3.5)
Leg splits forward, left leg to ground (cm)	37.7	(3.2)	38.4	(3.0)	39.4	(3.3)[a]	38.4	(3.6)	38.4	(3.6)	38.8	(3.2)	38.5	(3.5)	38.7	(3.5)
Bending backward (deg)	56.3	(6.9)	57.1	(6.4)	55.1	(7.9)	54.3	(7.9)	55.2	(8.4)	57.1	(8.8)	58.4	(7.9)	50.7	(10.4)[c]
Leg splits sideways, symphysis to ground (cm)	48.4	(10.5)	51.5	(7.9)	54.5	(8.9)[a]	49.8	(8.6)	51.1	(8.9)	51.2	(8.8)	51.0	(8.3)	52.7	(8.3)
One-legged knee bend, right symphysis to ground (cm)	31.1	(3.9)	31.4	(3.5)	31.2	(3.7)	31.3	(3.4)	30.5	(3.5)	31.4	(3.2)	30.8	(3.1)	31.1	(3.2)
One-legged knee bend, left symphysis to ground (cm)	31.0	(3.6)	31.0	(3.1)	32.4	(3.3)	31.5	(3.2)	30.9	(3.6)	31.5	(3.3)	30.9	(3.1)	31.4	(3.1)
Forward trunk bending, fingertips to ground (cm)	−5.6	(7.7)	−7.7	(5.1)	−0.9	(8.5)[a]	−4.6	(7.7)	−5.8	(6.9)	−1.5	(7.9)[c]	−4.1	(8.7)	−0.1	(7.8)[a]

[a] Significance of difference between high- and low-level players within each age group after correction for height: $P \leq 0.05$.
[b] Significance of difference between high- and low-level players within each age group after correction for height: $P \leq 0.01$.
[c] Significance of difference between high- and low-level players within each age group after correction for height: $P \leq 0.001$.

Figure 4- Flexibility test

After compliance with the tests, which required 180 minutes to complete, was excellent in all subgroups. **The comprehensive information given to the teams before the tests, the relevance of the tests to the footballers' normal activity, as well as the complete dedication of the examiners, were probably the most important factors motivating the players and allowed for the generation of reliable test results.**

Mean values in different age and skill-level groups were established for tests of flexibility, football skills, power, speed, and endurance. In comparing different age and skill levels, we found no systematic differences for flexibility or for accuracy in

Shooting or heading into the goal. Forward trunk bending was the only flexibility test that revealed (after correction for body height) differences between high- and low-level groups. Among the football skill tests, speed dribbling was the most powerful in discriminating between age and skill levels. In youth players, dexterity tests (juggling, passing) differed significantly between high- and low-level groups.

The Result:

The significant differences between the qualities of the warm-up performed by the different skill-level groups possibly reflected the lower standard of training in less skilled players.

In summary, the F-MARC test battery proved to be a feasible instrument for **assessing physical performance as well as football-specific skills**. **The results have not been systematically checked for their reproducibility because of the complexity of the examination**.

However, they are presented as a **database for comparative purposes in future studies**, such as in the evaluation of different sociocultural environments on performance and the incidence of injuries.

The present study strongly supports the analysis of an individual player's physical Profile, in relation to mean values for a similar age group and skill level, might be of assistance to the coach in objectively evaluating the effects of a specific training program. It may also be of use to the

responsible physician and physical therapist in monitoring progress during rehabilitation after football injuries.

2- Sports Analytics for Football League Table and Player Performance Prediction

By:
Victor Chazan – Pantzalis
The Data Mining and Analytics research group School of Science and Technology International Hellenic University Thermi, Greece

Christos Tjortjis the Data Mining and Analytics research group School of Science and Technology International Hellenic University Thermi, Greece

This research focuses on statistics from the previous season and historical data. Also, some financial data (i.e. transfer spending, team salaries) are exploited to contribute to the team evaluation process. The novelty of this research is that advanced metrics were used, such
As xG and Pezzali score to predict next season performance before the season begins, not after matches have already been played and recorded. Additionally, attackers are usually graded higher than defenders, even if they are not always more influential in team strategy. So, regarding player evaluation, this research attempts to identify skills and features suitable that make good defenders.
The flow of events taking place before we can get any meaningful experimental results, as well as the way the data were acquired and their preprocessing. The block diagram which summarizes the process

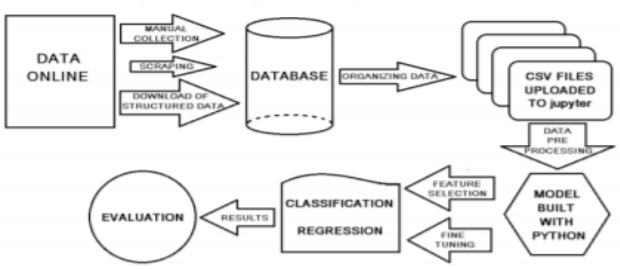


Figure 5-Process

Steps:

1- At first, the appropriate data had to be found

2- A free database had been downloaded and used for the experiments.

3- It demonstrates player ratings for several football skills.

4- Players are rated by domain experts.

5- The database was split into different csv files, according to what data was essential for each experiment.

6- The csv files were uploaded to jupyter.

7- They were checked for null values, duplicates, noise etc.

8- Python was used to clean the data and build the models.

9-data transformation and data reduction took place to keep only the appropriate features for each classification or regression.

10- Results were evaluated in terms of accuracy, error rates and bias involved.

This research paper discussed 2 experiments:

1st experiment: Team Performance Prediction

2nd experiment: Player Performance Prediction

This experiment focuses on individual players, specifically central defenders. In rating systems,

The purpose of this research is to examine the characteristics and the statistics for central defenders in comparison with their season rating band and decide which of them contribute more to distinguish a central defender as a top class player.

While it is easy to rate attacking players, according to the goals, key passes and assists, it is not straightforward what makes a good central defender. Consequently, there is very limited research on central defenders

The database was narrowed down to 59 players, as only central defenders, playing in English Premier League and having participated in at least 10 league matches for the 2016–17

Seasons were selected.

The main focus was on statistics regarding defensive player actions, but also, some team statistics were collected; despite demanding to build a model based on player performance, it must be acknowledged that a footballer's team has an impact both on his statistics but also on his overall rating.

First Approach is:

1- To normalize every numeric value of the dataset, so every attribute's range was transformed to range 0 to 1.

2- Multiple regression models were built with every possible feature.

3-it’s noticed that interceptions are the most important characteristic, followed by team overall rating, as expected. Players’ best attributes turned out to be their jumping reach, versatility, acceleration and first touch on The ball.

Another approach that was followed was to split the dataset’s features Into three categories:

1. Player characteristics and attributes.
2. Player statistics.
3. Team statistics.

The target was to build three multiple linear regression models (i.e. player attributes based, and statistics based), but with fewer independent variables than in the first approach.
The final model was to build with seven features, which seem to be the most influential for a Central defender.

The five assumptions of linear regression were also verified for this model; there was an indication of linearity in the model. Also, the expectation (mean) of residuals was almost zero and it appeared that there was (perfect) multicollinearity between features.

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.867
Model:	OLS	**Adj. R-squared:**	0.833
Method:	Least Squares	**F-statistic:**	25.03
Date:	Mon, 11 Nov 2019	**Prob (F-statistic):**	3.39e-16
Time:	02:24:20	**Log-Likelihood:**	64.550
No. Observations:	59	**AIC:**	-103.1
Df Residuals:	46	**BIC:**	-76.09
Df Model:	12		
Covariance Type:	nonrobust		

Omnibus:	1.400	**Durbin-Watson:**	1.912
Prob(Omnibus):	0.496	**Jarque-Bera (JB):**	1.161
Skew:	-0.136	**Prob(JB):**	0.560
Kurtosis:	2.369	**Cond. No.**	1.41e+05

	coef	std err	t	P>|t|	[0.025	0.975]
const	6.9602	0.142	49.122	0.000	6.675	7.245
x1	-0.0327	0.004	-7.776	0.000	-0.041	-0.024
x2	0.0032	0.001	5.650	0.000	0.002	0.004
x3	-0.0113	0.002	-4.635	0.000	-0.016	-0.006
x4	6.782e-05	2.04e-05	3.317	0.002	2.67e-05	0.000
x5	0.2326	0.039	6.017	0.000	0.155	0.310
x6	0.0972	0.028	3.525	0.001	0.042	0.153
x7	0.1261	0.021	6.127	0.000	0.085	0.168
x8	-0.1559	0.042	-3.749	0.000	-0.240	-0.072
x9	-0.0481	0.019	-2.588	0.013	-0.086	-0.011
x10	0.7259	0.124	5.862	0.000	0.477	0.975
x11	0.9718	0.226	4.292	0.000	0.516	1.428
x12	2.1514	0.743	2.896	0.006	0.656	3.647

Figure 6- Regression results

Additionally, all five assumptions of linear regression were met; Linearity of the model was obvious

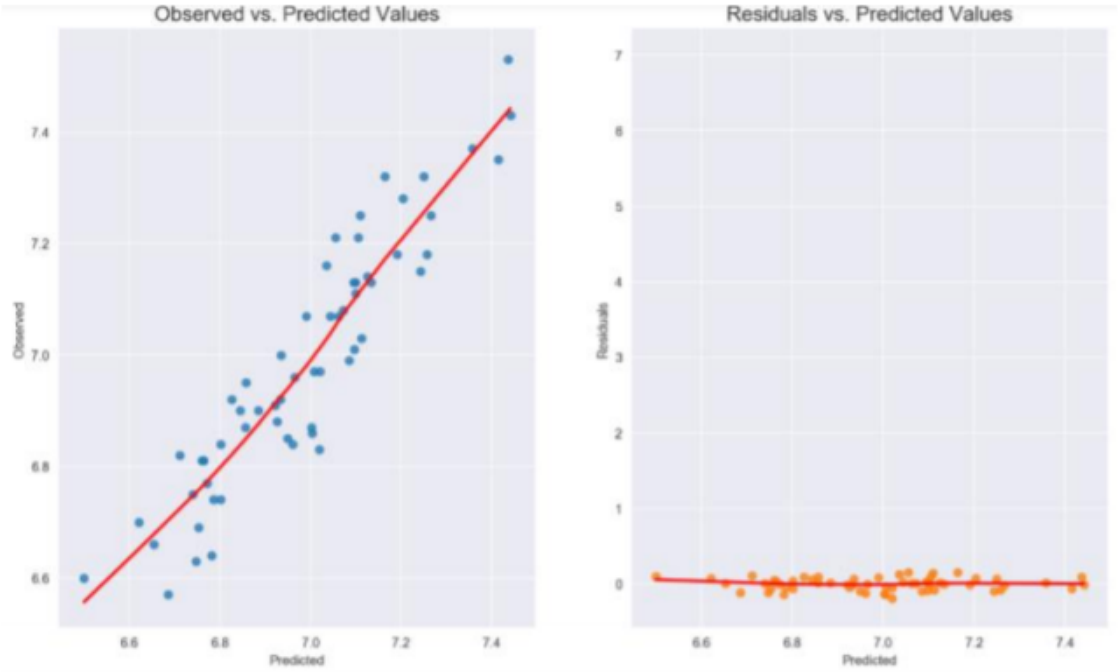


Figure 5: *Linearity of the second model.*

Figure 7- Linearity of model

The third set of features (i.e. team statistics) did not help to build a satisfactory model. There was an indication that the only of those variables worth noting is “Team Rating”.

The results of all models deployed leads to the most critical attributes and game actions for predicting the performance of a central defender can be described in the following
List. It must be highlighted that attacking skills are not absent from the list, following the way modern defenders are expected to play:
Interceptions, Clearances, Aerials Won, Tackles, Jumping reach, Versatility, Acceleration,
First touch on ball, Age, Passing, Vision, Determination, Strength, Professionalism and ability to perform well in important matches, International Caps, Minutes Played, Fouls, Inaccurate short passes, Key passes, Goals, Team’s rating.

Model Challenges:

1- The database used consisted of defenders based only in England, because it would be inefficient to include players from different leagues.

2- Football matches are not affected only by team ability and player skills. There are some external factors that cannot be predicted. Luck is an imponderable factor. Long term injuries of
Important players are also part of the game. “Strange” results in matches where one or both teams are not in real need of victory are often observed.

Model Achievements:

The main achievement was locating a set of attributes and skills that a central defender must improve in order to be considered a top class player. Of course, every player is different and has his own playing style, but it would be very useful for coaches to have a specific target when training a player.

3- Article Evaluation of the Technical Performance of Football Players in the UEFA Champions League

By: International Journal of Environmental Research and Public Health - Jan 2020

This study aimed to assess the technical match performance of top-class football players in

A long-term perspective. Technical performance profiles of players according to five playing positions (central defender, full back, wide midfielder, central midfielder, forward) and five situational variables (competition stage, match location, quality of team, quality of opponent, match outcome) were established. Technical match data of players in the UEFA Champions League from season 2009–2010 to 2016–2017 were analyzed.

(1000 matches, 768 matches from group stage, 232 matches from knockout stage) were collected for analysis. Match data were obtained from a public-accessed football statistics website “whoscored.com”

Also it's considered that goalkeepers have a very limited applicability of technical variables to the performance, match data for this position was excluded from the sample.

All players in the sample were divided into 5 groups according to their playing positions

Central defender

Full back

Wide midfielder

Central midfielder

Forward

Twenty-five technical performance-related match actions or events were chosen as variables in the present study and were divided into four groups

Goal scoring: Shot, Shot on target.

Attacking: Dispossessed, Unsuccessful touch, Fouled, Aerial won, Dribble, Offside.

Defending: Yellow card, Total tackle, Interception, Clearance, Blocked shot, foul

Passing and organizing : Assist, Touch, Key pass, Pass accuracy (%), Pass, Cross, Accurate cross, Long ball, Accurate long ball, Through ball, Accurate through ball.

Five situational variables:

(1) Competition stage: group stage and knockout stage;

(2) Match location: home and away;

(3) Quality of team: teams that qualified into the knockout stage and teams that didn't qualify into the knockout stage;

(4) Quality of opponent: opponents that qualified into the knockout stage and opponents that didn't qualify into the knockout stage;

(5) Match outcome: win, draw and lose

Table 2. Statistical differences of players' match performance across five playing positions and five competing situations.

Position	Group-Knockout			Home-Away			Non-Qualified-Qualified			Non-Qualified Opp.-Qualified Opp.			Draw/Lose-Win		
	Variable	Effect Size	Inference	Variable	Effect Size	Inference	Variable	Effect Size	Inference	Variable	Effect Size	Inference	Variable	Effect Size	Inference
CD				Clearance	0.22 ± 0.06	S *	Touch	0.47 ± 0.06	S ****	Touch	−0.39 ± 0.06	S ****	Touch	0.31 ± 0.07	S ****
				Pass	−0.22 ± 0.06	S *	Pass	0.50 ± 0.06	S ****	Pass	−0.39 ± 0.06	S ****	Pass	0.33 ± 0.07	S ****
							PA	0.29 ± 0.06	S ***	AW	−0.26 ± 0.06	S **			
							AccLB	0.21 ± 0.06	S *						
FB				Clearance	0.23 ± 0.07	S *	Touch	0.44 ± 0.07	S ****	Touch	−0.35 ± 0.07	S ****	Assist	0.39 ± 0.07	S ****
				Cross	−0.22 ± 0.07	S *	PA	0.32 ± 0.07	S ****	Pass	−0.38 ± 0.07	S ****	Touch	0.32 ± 0.07	S ****
							Pass	0.50 ± 0.07	S ****	AccLB	−0.22 ± 0.07	S *	Pass	0.39 ± 0.07	S ****
													PA	0.27 ± 0.07	S ***
WM	Foul	0.21 ± 0.11	S *	Shot	−0.26 ± 0.10	S **	Touch	0.59 ± 0.10	S ****	Touch	−0.42 ± 0.10	S ****	Assist	0.47 ± 0.12	S ****
				KP	−0.28 ± 0.10	S **	Pass	0.56 ± 0.10	S ****	Pass	−0.40 ± 0.10	S ****	Pass	0.51 ± 0.11	S ****
				Cross	−0.21 ± 0.10	S *	ThB	0.35 ± 0.10	S ***	Shot	−0.26 ± 0.10	S **	ShotOT	0.38 ± 0.11	S ****
				AccCross	−0.21 ± 0.10	S *	PA	0.30 ± 0.10	S ***	KP	−0.24 ± 0.10	S **	Touch	0.55 ± 0.11	S ****
							AccThB	0.26 ± 0.10	S **	ShotOT	−0.24 ± 0.10	S *	Shot	0.31 ± 0.11	S ***
							Shot	0.24 ± 0.10	S **	PA	−0.21 ± 0.10	S *	PA	0.31 ± 0.11	S ***
							ShotOT	0.29 ± 0.10	S **				KP	0.29 ± 0.11	S **
							Assist	0.21 ± 0.10	S *				ThB	0.27 ± 0.11	S **
							KP	0.21 ± 0.10	S *				AccThB	0.26 ± 0.11	S **
							AccLB	0.20 ± 0.10	S *				AccLB	0.23 ± 0.11	S *
CM				Shot	−0.20 ± 0.06	S *	Touch	0.49 ± 0.06	S ****	Touch	−0.39 ± 0.06	S ****	Assist	0.33 ± 0.07	S ****
							PA	0.35 ± 0.06	S ****	Pass	−0.37 ± 0.06	S ****	Touch	0.39 ± 0.06	S ****
							Pass	0.49 ± 0.06	S ****	LB	−0.23 ± 0.06	S ***	Pass	0.38 ± 0.06	S ****
							ThB	0.24 ± 0.06	S **				PA	0.21 ± 0.06	S *
							Assist	0.21 ± 0.06	S *						
FW				Shot	−0.32 ± 0.10	S ***	Touch	0.47 ± 0.10	S ****	Shot	−0.36 ± 0.10	S ****	Assist	0.46 ± 0.11	S ****
				KP	−0.28 ± 0.10	S **	Pass	0.43 ± 0.10	S ****	ShotOT	−0.32 ± 0.10	S ***	Shot	0.52 ± 0.10	S ****
				ShotOT	−0.22 ± 0.10	S *	Shot	0.32 ± 0.10	S ***	KP	−0.35 ± 0.10	S ***	ShotOT	0.57 ± 0.10	S ****
				AccCross	−0.23 ± 0.10	S *	ShotOT	0.32 ± 0.10	S ***	Assist	−0.28 ± 0.10	S **	Touch	0.46 ± 0.10	S ****
							Assist	0.28 ± 0.10	S **	Touch	−0.28 ± 0.10	S **	Pass	0.42 ± 0.10	S ****
							KP	0.30 ± 0.10	S **	Pass	−0.26 ± 0.10	S **	ThB	0.37 ± 0.11	S ****
							PA	0.28 ± 0.10	S **	ThB	−0.28 ± 0.10	S **	KP	0.34 ± 0.10	S ***
							Dribble	0.21 ± 0.10	S *	AccThB	−0.27 ± 0.10	S **	AccThB	0.33 ± 0.11	S ***
													Dribble	0.28 ± 0.10	S **

Note: Effect sizes are presented as the magnitude of the true difference in means ± 90% confidence interval, only the variables that showed clear differences were included. Positive effect size indicates that the mean values of variables from group A bigger than the mean values of variables from group B, negative effect size indicates that the mean values of variables from group B bigger than the mean values of variables from group A, e.g., group B-group A: group stage-knockout stage. Letters in parentheses denote the magnitude: t = trivial; s = small. Asterisks indicate the likelihood for the magnitude of the true difference in means as follows: * possible; ** likely; *** very likely; **** most likely. Abbreviations: AccCross = accurate cross pass; AccLB = accurate long ball; AccThB = accurate through; AW = aerial won; CD = central defender; CM = central midfielder; FB = full back; FW = forward; KP = key pass; PA = pass accuracy in %; ShotOT = shot on target; ThB = through ball; ball WM = wide midfielder.

Figure 8- Features of positions

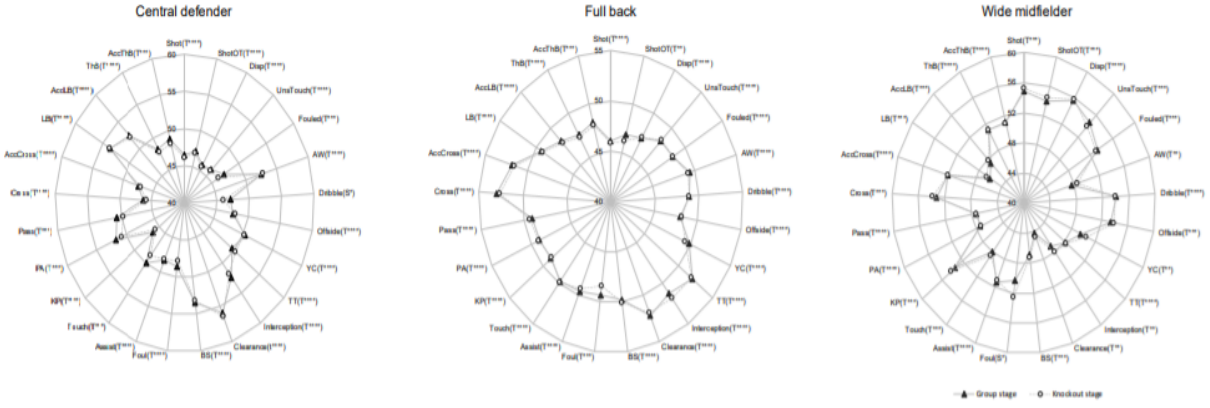


Figure 9- Features of 3 positions

25 technical performance-related actions or events of players were transformed into standardized scores (Z-Score) and were unified into the same scale by the formula T = 10Z + 50

There is a new finding concerning the influence of competition stage on player performance that did not appear in the previous research due to the investigation of domestic matches

Another important result that can be found is touches and passes are the only two variables that showed clear differences for players of all positions when taking quality of team, quality of opponent, and match outcome into account

-all variables showed small or trivial differences across five situational variables

-analysis of long-term data may reduce the impact of situational variables on players' performance due to the higher stability of the performances during eight seasons

- The differences of player's technical performance could mainly be identified in variables related to passing and organizing, especially touches and passes

- Three of the situational variables (quality of team, quality of opponent, and match outcome) had a relatively greater influence on players' technical performance than competition stage and match location

- strong teams are more likely to adopt a possession-based playing style, or that improving players' ability in the aspect of passing and organizing can help to achieve a better match performance for football teams in the group

stage. Moreover, forwards' performance in shots, shots on target, and dribbles showed clear differences within these three contexts, which might be a result of the differences of skill level between forwards and the support they got from midfielders based on the advantage of possession

- Touches and passes were the only two variables in which match performance of players from all playing positions showed clear differences when considering the effect of quality of team, quality of opponent, and match outcome. This provides us with the opportunity to explore the interaction between five playing positions and three situational variables

This study contributes to the current research on performance analysis By establishing more comprehensive and detailed technical profiles to examine the
Interaction of positional and situational variables on players.

Analysis and Requirements

1. Requirements:

A. Functional Requirements

REQ-x	Requirement
REQ1	The system User can search for players inside the database of the API based on his name or by teams, and display all the information and statistics of the player.
REQ2	The system user (Coach, Scouting team) can get a prediction of performance of the selected player based on the player statistics.
REQ3	The system user (Coach) can add statistics of the player of youth team or unknown players, and rate his performance.
REQ4	The system user (Data analyst) can get data of the selected player of multiple seasons.
REQ5	The System extracts data from a video of the match and gives a detailed report by the application.
REQ6	The System user (Player) can monitor his statistics.
REQ7	The Data API Should Provide Data For the Application.

B. Non Functional Requirements

REQ-x	Requirement
REQ8	Authorized access to the user is provided as a coach wants to get the prediction of any player.
REQ9	Authorized access to the user is provided as a scouter wants to hunt the underrated players.
REQ10	Authorize access to the player to follow up his performance.
REQ11	Authorize access to the data analyst to visualize the data.

2. FUNCTIONAL Requirement Specification:

A. Stakeholders

- Coach
- Scouting Team
- Player
- Data Analyst
- Mobile Application Developer
- Football Fan

B. Actors and Goals

- Coach:
    - Search for players Data to see potential player admission.
    - Get Prediction on player performance.
    - Insert unregistered player data to get a rating that reflects the player performance.

- Scouting Team:
    - Search for players Data to see potential player admission.
    - Get Prediction on player performance.

- Player:
    - Can Search for his own numbers to see his performance and try to be better and overcome his drawbacks.

- Data Analyst:
    - Search for players Data to analyze them.
    - Access our application Data to make his own analysis.

- Mobile Application Developer:
    - Develop the application so the above stakeholders can interact with the application
- Video Analysis System:
    - Analyze matches videos and upload the player's analysis to the database.
- Data API:
    - Provide historical data for the android application

C. Use Case

I. Use Case Description

Use Case	Name	Description	Requirement
UC-1	Search for player data	To manage a player, coach, scout or analyst to search for a specific player data.	REQ3
UC-2	Get Prediction	To manage coach or scout to get predictions about future performance of players	REQ4
UC-3	Rate unregistered player	Coach can add statistics of an unknown player, and rate his performance.	REQ5
UC-4	Extract Data from video	System can Automatically extract parameters from the football video.	REQ7
UC-5	Upload Data	After getting the data from the video the system uploaded it to the database.	REQ7
UC-6	Provide data	The Data API provides the app with the desired Data	REQ9

ii. Use Case Diagram:

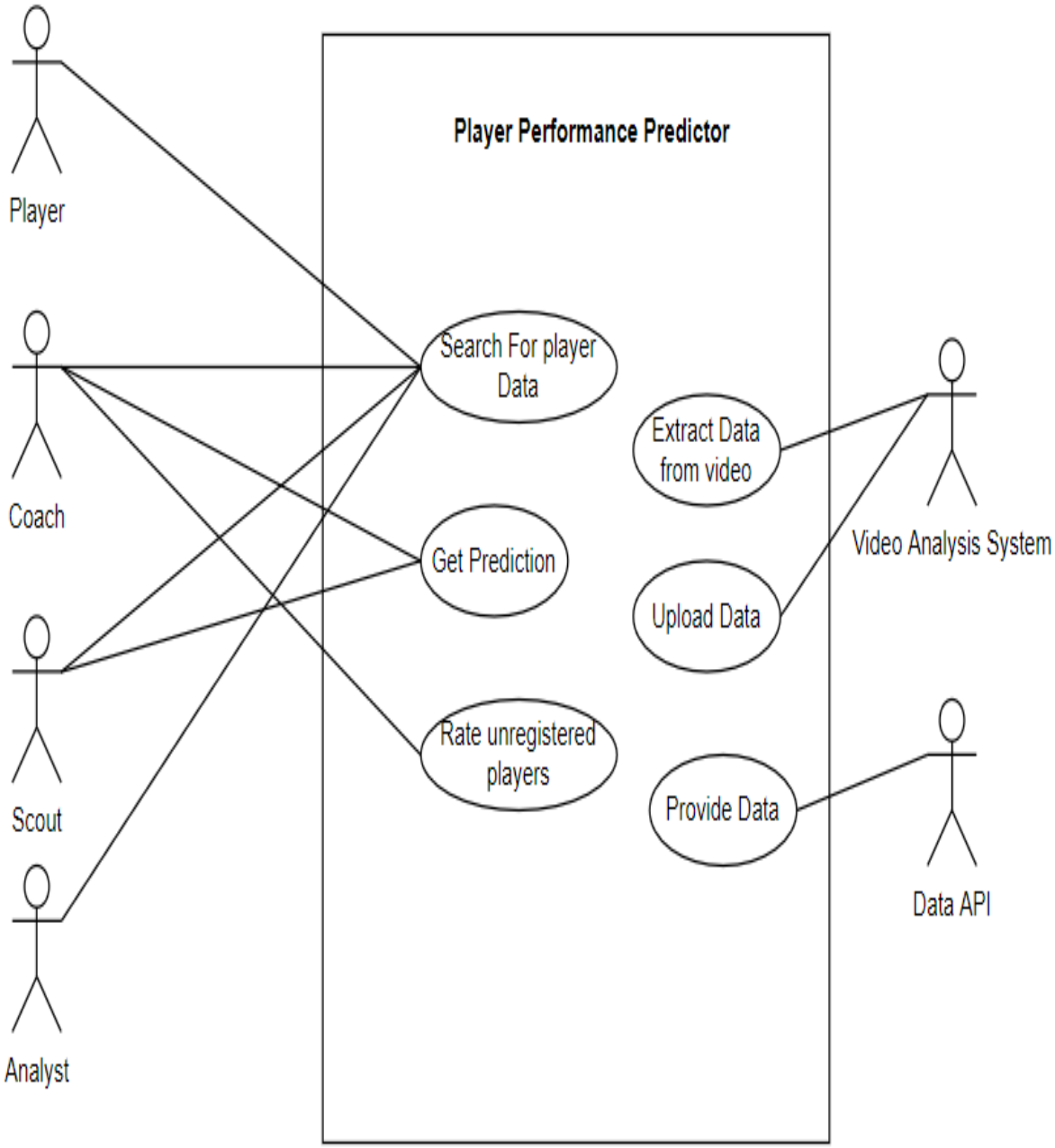


Figure 10- Use case

D. System Sequence Diagram

All users search for player data

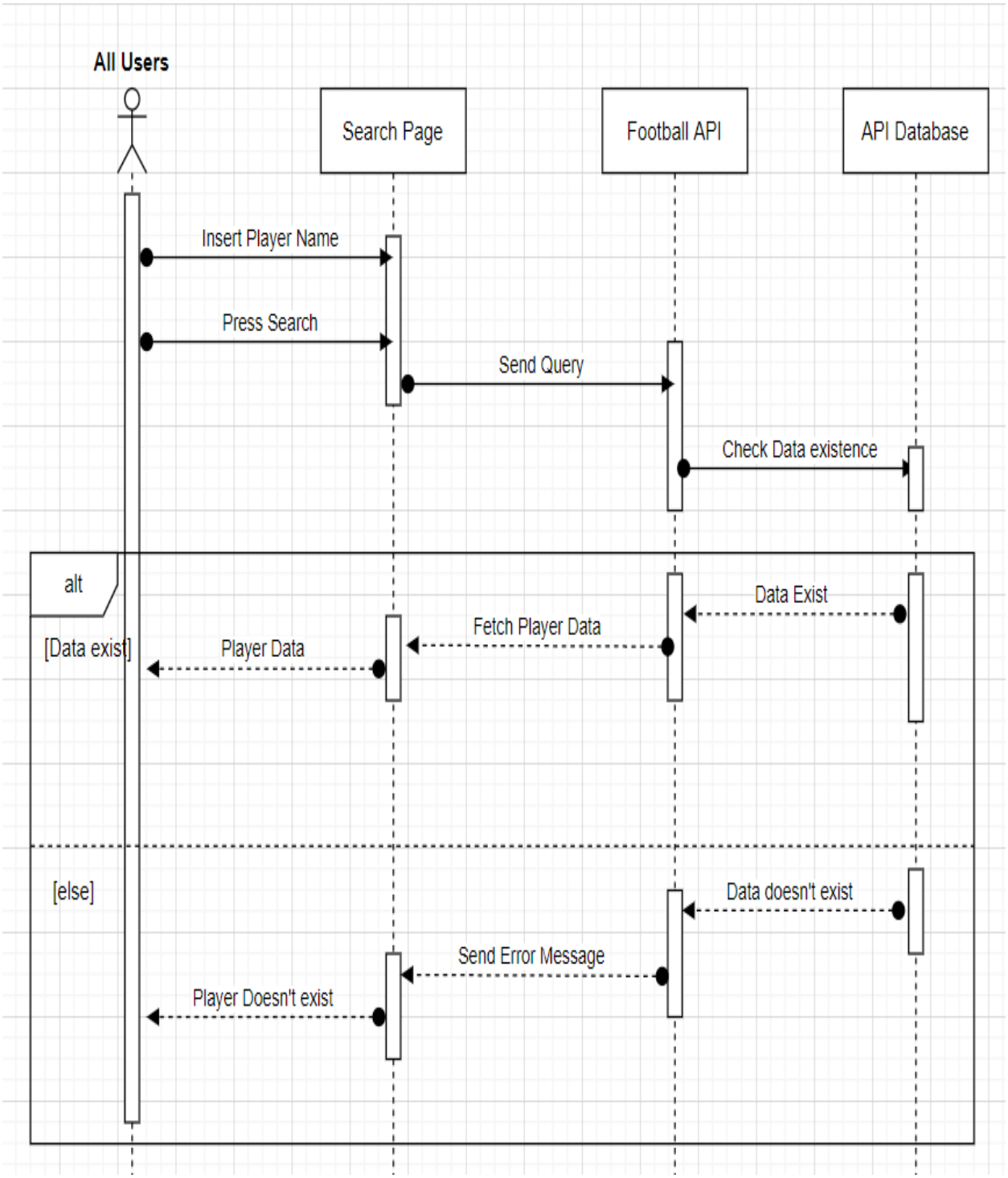


Figure 11- sequence diagram1

Coach and scout get performance prediction

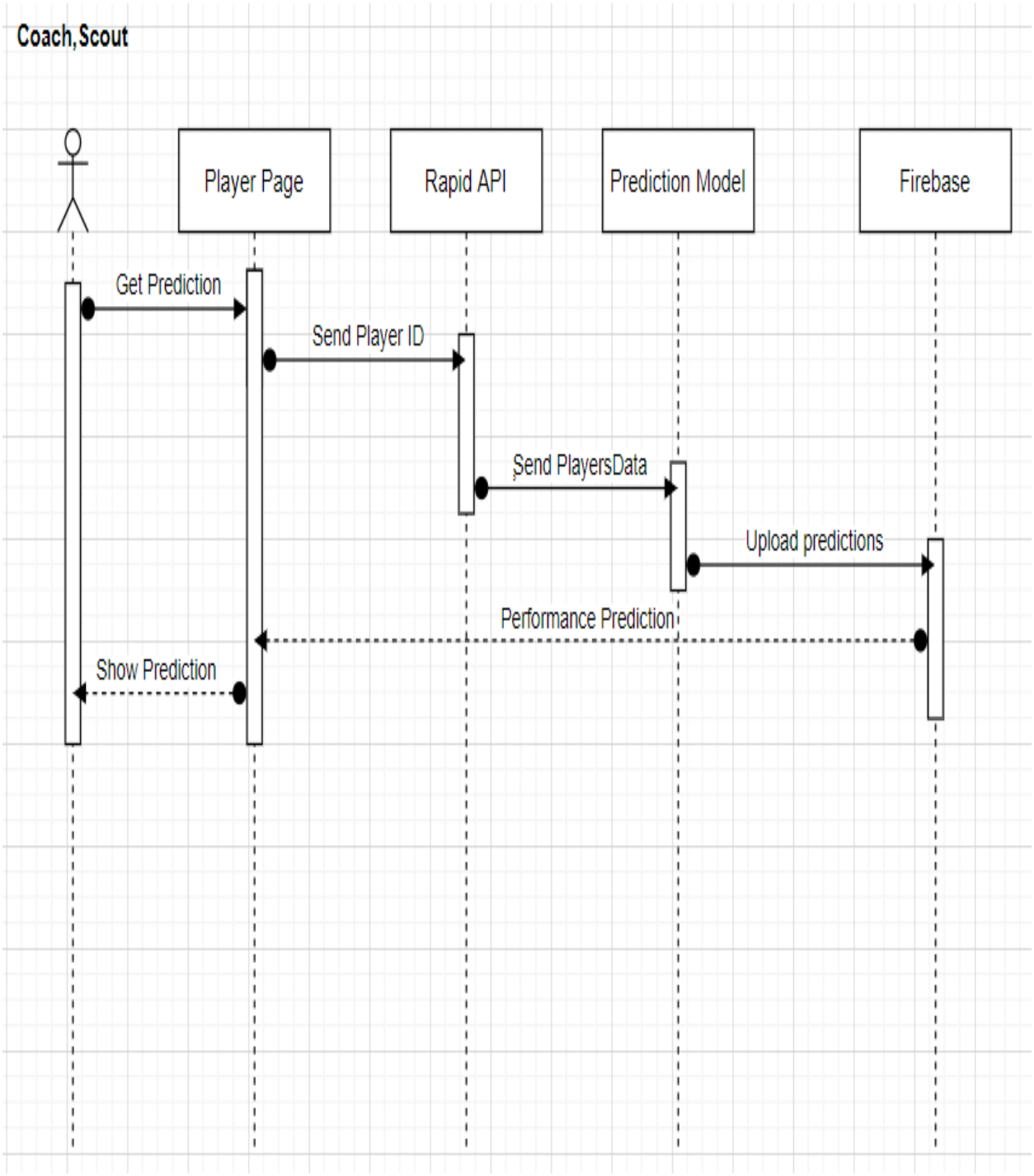


Figure 12- sequence diagram2

Coach gets player rating

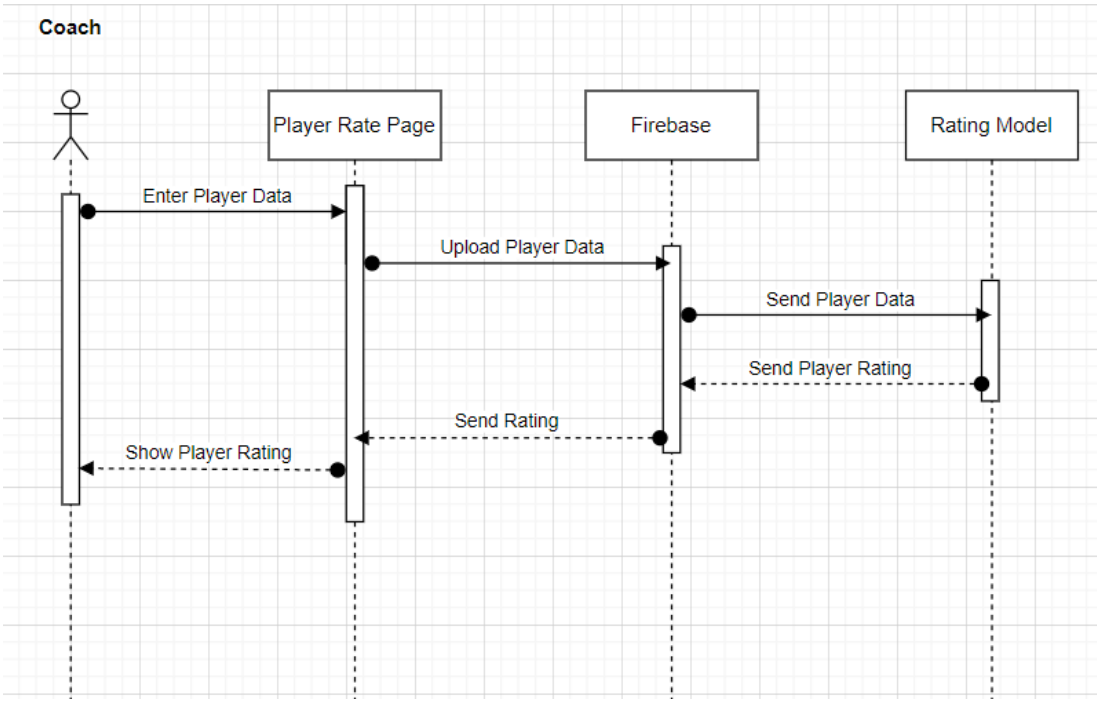


Figure 13- sequence diagram3

E. Class Diagram

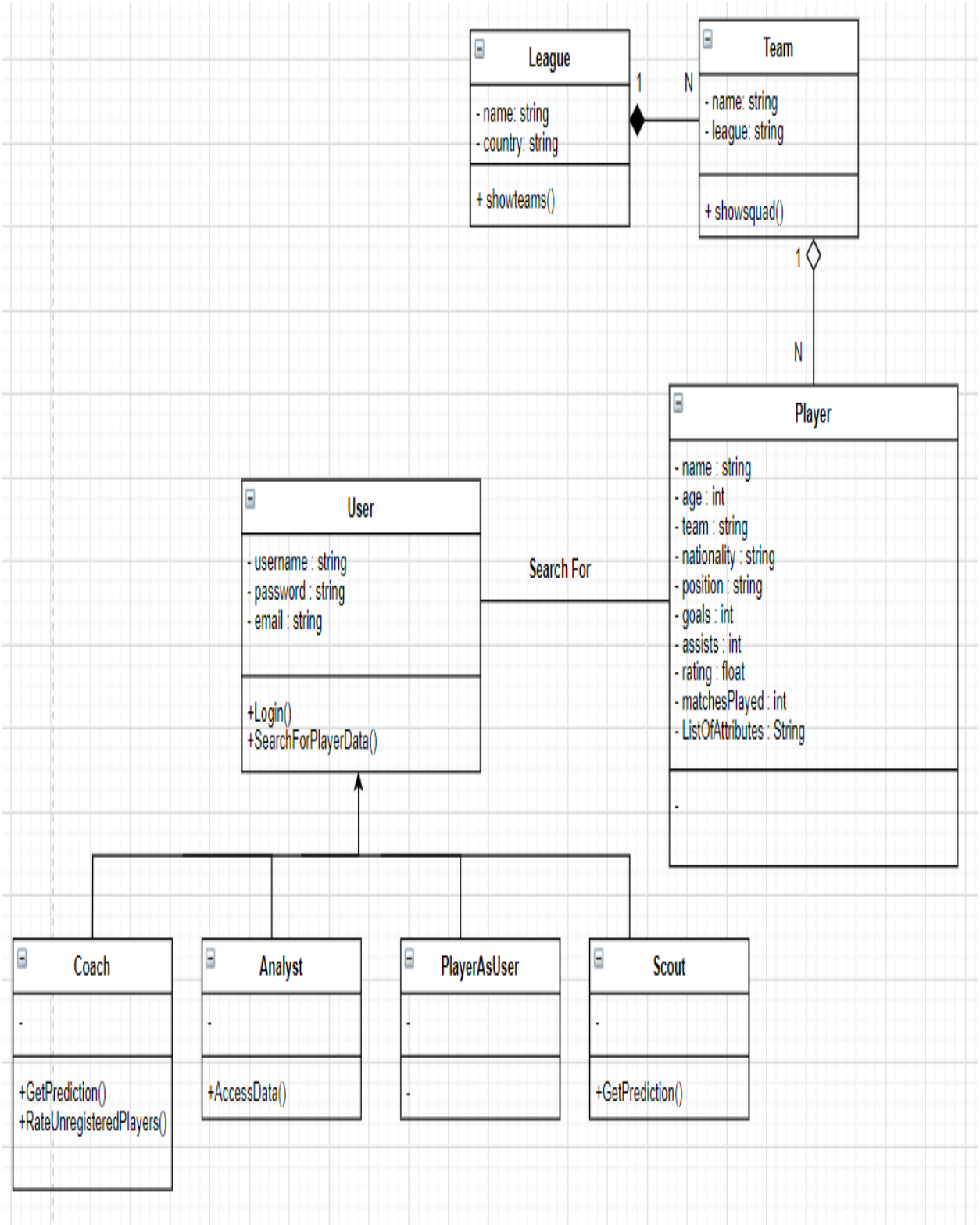


Figure 14- class diagram

IMPLEMENTATION

Dataset and Features

Data Analysis

Data analysis is a process of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, informing conclusions, and supporting decision-making.

Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientifically and helping businesses operate more effectively.

There are many data analysis techniques but our project aims to use **Predictive Analytics** technique that focuses on the application of statistical models for predictive forecasting or classification.

To apply the process of Data Analysis we have to make 3 main steps

- Gathering Data
- Cleaning Data
- Assess Data

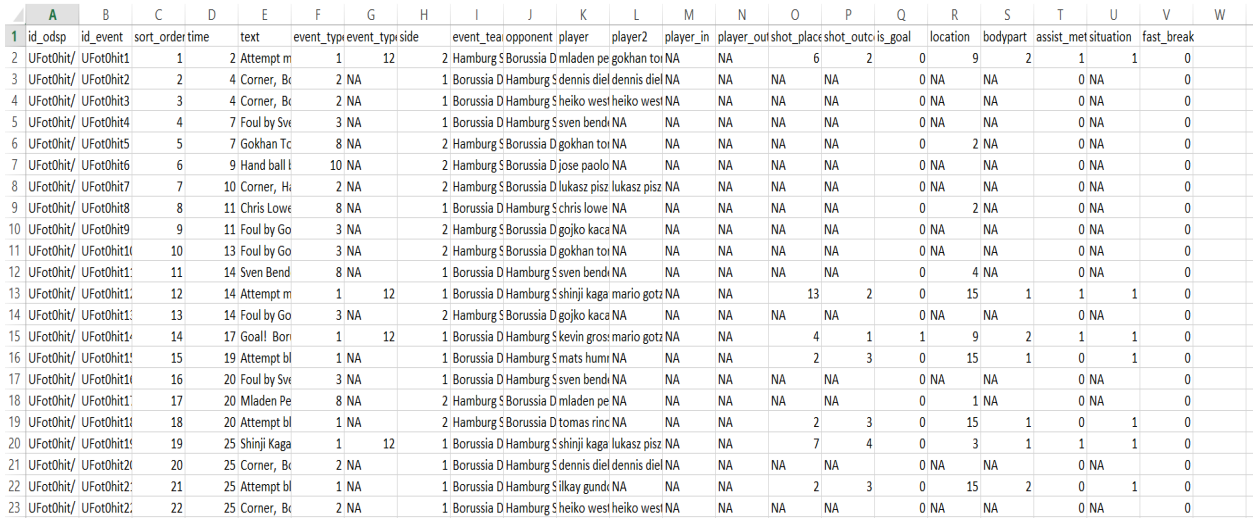
Gathering Data

Our project Data is provided by an **API** that provides historical data of previous football matches the data provider is **Football API** powered by **Rapid API** That has data of many leagues like English Premier League, Serie

A, Ligue 1 and Bundesliga and we succeeded to gather data from English Premier League from the last 5 seasons.

We followed some steps to extract the data:

At the beginning the extracted data was represented as events and ids



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	id_odsp	id_event	sort_order	time	text	event_type	event_type	side	event_tea	opponent	player	player2	player_in	player_out	shot_place	shot_outc	is_goal	location	bodypart	assist_met	situation	fast_break	
2	UFot0hit/	UFot0hit1	1	2	Attempt m	1	12	2	Hamburg S	Borussia D	mladen pe	gokhan tor	NA	NA	6	2	0	9	2	1	1	0	
3	UFot0hit/	UFot0hit2	2	4	Corner, Bo	2	NA	1	Borussia D	Hamburg S	dennis diel	dennis diel	NA	NA	NA	NA	0	NA	NA	0	NA	0	
4	UFot0hit/	UFot0hit3	3	4	Corner, Bo	2	NA	1	Borussia D	Hamburg S	heiko west	heiko west	NA	NA	NA	NA	0	NA	NA	0	NA	0	
5	UFot0hit/	UFot0hit4	4	7	Foul by Sv	3	NA	1	Borussia D	Hamburg S	sven bend	NA	NA	NA	NA	NA	0	NA	NA	0	NA	0	
6	UFot0hit/	UFot0hit5	5	7	Gokhan To	8	NA	2	Hamburg S	Borussia D	gokhan tor	NA	NA	NA	NA	NA	0	2	NA	0	NA	0	
7	UFot0hit/	UFot0hit6	6	9	Hand ball b	10	NA	2	Hamburg S	Borussia D	jose paolo	NA	NA	NA	NA	NA	0	NA	NA	0	NA	0	
8	UFot0hit/	UFot0hit7	7	10	Corner, Ha	2	NA	2	Hamburg S	Borussia D	lukasz pisz	lukasz pisz	NA	NA	NA	NA	0	NA	NA	0	NA	0	
9	UFot0hit/	UFot0hit8	8	11	Chris Lowe	8	NA	1	Borussia D	Hamburg S	chris lowe	NA	NA	NA	NA	NA	0	2	NA	0	NA	0	
10	UFot0hit/	UFot0hit9	9	11	Foul by Go	3	NA	2	Hamburg S	Borussia D	gojko kaca	NA	NA	NA	NA	NA	0	NA	NA	0	NA	0	
11	UFot0hit/	UFot0hit1(	10	13	Foul by Go	3	NA	2	Hamburg S	Borussia D	gokhan tor	NA	NA	NA	NA	NA	0	NA	NA	0	NA	0	
12	UFot0hit/	UFot0hit1	11	14	Sven Bend	8	NA	1	Borussia D	Hamburg S	sven bend	NA	NA	NA	NA	NA	0	4	NA	0	NA	0	
13	UFot0hit/	UFot0hit1	12	14	Attempt m	1	12	1	Borussia D	Hamburg S	shinji kaga	mario gotz	NA	NA	13	2	0	15	1	1	1	0	
14	UFot0hit/	UFot0hit1	13	14	Foul by Go	3	NA	2	Hamburg S	Borussia D	gojko kaca	NA	NA	NA	NA	NA	0	NA	NA	0	NA	0	
15	UFot0hit/	UFot0hit1	14	17	Goal! Bor	1	12	1	Borussia D	Hamburg S	kevin gros	mario gotz	NA	NA	4	1	1	9	2	1	1	0	
16	UFot0hit/	UFot0hit1	15	19	Attempt bl	1	NA	1	Borussia D	Hamburg S	mats hum	NA	NA	NA	2	3	0	15	1	0	1	0	
17	UFot0hit/	UFot0hit1(	16	20	Foul by Sv	3	NA	1	Borussia D	Hamburg S	sven bend	NA	NA	NA	NA	NA	0	NA	NA	0	NA	0	
18	UFot0hit/	UFot0hit1	17	20	Mladen Pe	8	NA	2	Hamburg S	Borussia D	mladen pe	NA	NA	NA	NA	NA	0	1	NA	0	NA	0	
19	UFot0hit/	UFot0hit1(	18	20	Attempt bl	1	NA	2	Hamburg S	Borussia D	tomas rinc	NA	NA	NA	2	3	0	15	1	0	1	0	
20	UFot0hit/	UFot0hit1	19	25	Shinji Kaga	1	12	1	Borussia D	Hamburg S	shinji kaga	lukasz pisz	NA	NA	7	4	0	3	1	1	1	0	
21	UFot0hit/	UFot0hit2(	20	25	Corner, Bo	2	NA	1	Borussia D	Hamburg S	dennis diel	dennis diel	NA	NA	NA	NA	0	NA	NA	0	NA	0	
22	UFot0hit/	UFot0hit2	21	25	Attempt bl	1	NA	1	Borussia D	Hamburg S	ilkay gund	NA	NA	NA	2	3	0	15	2	0	1	0	
23	UFot0hit/	UFot0hit2	22	25	Corner, Bo	2	NA	1	Borussia D	Hamburg S	heiko west	heiko west	NA	NA	NA	NA	0	NA	NA	0	NA	0	

Figure 15- events

We extracted the data of players in different league and our main focus was on the premier league.

Then we get each index of the events and leagues

```
event_type
0          Announcement
1          Attempt
2          Corner
3          Foul
4          Yellow card
5          Second yellow card
6          Red card
7          Substitution
8          Free kick won
9          Offside
10         Hand ball
11         Penalty conceded

event_type2
12         Key Pass
13         Failed through ball
14         Sending off
15         Own goal

side
1          Home
2          Away

shot_place
1          Bit too high
2          Blocked
3          Bottom left corner
4          Bottom right corner
5          Centre of the goal
6          High and wide
7          Hits the bar
8          Misses to the left
9          Misses to the right
10         Too high
11         Top centre of the goal
12         Top left corner
```

Figure 16-events types

api » results	api » leagues » leag	api » leagues » nam	api » leagues » type	api » leagues » cour	api » leagues » cour	api » leagues » seas	api » leagues » seas	api » leagues » seas	api » leagues » logo	api » leagues » flag	api » leagues » stan	api » leagues » is_c	api » leagues » cove	api » leagues
3029	1	World Cup	Cup	World		2018	2018-06-14	2018-07-15	https://media.api-sports.io/football/leagues		1	1	TRUE	TRUE
	2	Premier League	League	England	GB	2018	2018-08-10	2019-05-12	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	3	Championship	League	England	GB	2018	2018-08-03	2019-05-27	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	4	Ligue 1	League	France	FR	2018	2018-08-10	2019-05-24	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	5	Ligue 2	League	France	FR	2018	2018-07-27	2019-05-17	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	6	Serie A	League	Brazil	BR	2018	2018-04-14	2018-12-02	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	7	Serie B	League	Brazil	BR	2018	2018-04-13	2018-11-24	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	8	Bundesliga 1	League	Germany	DE	2018	2018-08-24	2019-05-27	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	9	Bundesliga 2	League	Germany	DE	2018	2018-08-03	2019-05-19	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	10	Eredivisie	League	Netherlands	NL	2018	2018-08-10	2019-05-28	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	11	Primeira Liga	League	Portugal	PT	2018	2018-08-10	2019-05-19	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	12	Liga de Honra	League	Portugal	PT	2018	2018-08-11	2019-05-19	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	13	J. League Div.1	League	Japan	JP	2018	2018-02-23	2018-12-01	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	14	J. League Div.2	League	Japan	JP	2018	2018-02-25	2018-12-02	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	15	Eliteserien	League	Norway	NO	2018	2018-03-11	2018-11-24	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	16	Ekstraklasa	League	Poland	PL	2018	2018-07-20	2019-05-19	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	17	Premier	League	Wales	GB	2018	2018-08-10	2019-05-19	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	18	Allsvenskan	League	Sweden	SE	2018	2018-04-01	2018-11-11	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	19	Vysshaya Liga	League	Belarus	BY	2018	2018-03-30	2018-12-02	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	20	Superligaen	League	Denmark	DK	2018	2018-07-13	2019-05-31	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	21	Eerste Divisie	League	Netherlands	NL	2018	2018-08-17	2019-05-03	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	22	Ligue 1	League	France	FR	2017	2017-08-04	2018-05-19	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	23	Ligue 1	League	France	FR	2016	2016-08-12	2017-05-20	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE
	24	Ligue 2	League	France	FR	2017	2017-07-28	2018-05-20	https://media.api-spo	https://media.api-spo	1	0	TRUE	TRUE

Figure 17- parameters

Then we matched the indexes with each event, league, parameter and player to get the data translated and linked

player_id	layer_nam	firstname	lastname	number	position	age	birth_date	irth_place	rth_count	ationality	height	weight	injured	rating	team_id	eam_nam	eague_id	league	season	captain	hots/tota	shots/on
1436	Petr u010	Petr	u010cech		Goalkeep	37	20/05/19	Plzeu014	Czech Rep	Czech Rep	196 cm	90 kg			42	Arsenal	524	Premier L	2019-202	0	0	0
138840	Robert Le	Robert Le	Burton		Midfielde	21	26/12/19	Gravesen	England	Wales	176 cm	69 kg			42	Arsenal	957	League Cı	2019-202	0	0	0
138781	Tolaji Bol	Tolaji	Bola		Defender	21	04/01/1999		England	England	188 cm	78 kg			42	Arsenal	957	League Cı	2019-202	0	0	0
167658	Matthew	Matthew	Smith		Defender	21	05/10/2000		England	England					42	Arsenal	524	Premier L	2019-202	0	0	0
167658	Matthew	Matthew	Smith		Defender	21	05/10/2000		England	England					42	Arsenal	1063	FA Cup	2019-202	0	0	0
1447	Nacho Mı	Ignacio	Monreal Eraso		Defender	35	26/02/19	Pamplon	Spain	Spain	179 cm	72 kg		6.733333	42	Arsenal	524	Premier L	2019-202	1	1	0
1447	Nacho Mı	Ignacio	Monreal Eraso		Defender	35	26/02/19	Pamplon	Spain	Spain	179 cm	72 kg			42	Arsenal	1063	FA Cup	2019-202	0	0	0
1462	L. Torreir	Lucas	Torreira Di Pascua		Midfielde	25	11/02/19	Fray Bent	Uruguay	Uruguay	168 cm	64 kg		6.77931	42	Arsenal	524	Premier L	2019-202	0	21	6
1462	L. Torreir	Lucas	Torreira Di Pascua		Midfielde	25	11/02/19	Fray Bent	Uruguay	Uruguay	168 cm	64 kg		7.25	42	Arsenal	957	League Cı	2019-202	0	5	1
1462	L. Torreir	Lucas	Torreira Di Pascua		Midfielde	25	11/02/19	Fray Bent	Uruguay	Uruguay	168 cm	64 kg		6.45	42	Arsenal	1063	FA Cup	2019-202	0	0	0
1462	L. Torreir	Lucas	Torreira Di Pascua		Midfielde	25	11/02/19	Fray Bent	Uruguay	Uruguay	168 cm	64 kg		6.816666	42	Arsenal	514	UEFA Eur	2019-202	0	7	1
19599	E. Martu	Damiu00	Martu00ednez		Goalkeep	29	02/09/19	Mar del P	Argentina	Argentina	195 cm	88 kg		7.244444	42	Arsenal	524	Premier L	2019-202	0	0	0
19599	E. Martu	Damiu00	Martu00ednez		Goalkeep	29	02/09/19	Mar del P	Argentina	Argentina	195 cm	88 kg		7.216666	42	Arsenal	1063	FA Cup	2019-202	0	0	0
19599	E. Martu	Damiu00	Martu00ednez		Goalkeep	29	02/09/19	Mar del P	Argentina	Argentina	195 cm	88 kg		5.95	42	Arsenal	957	League Cı	2019-202	0	0	0
19599	E. Martu	Damiu00	Martu00ednez		Goalkeep	29	02/09/19	Mar del P	Argentina	Argentina	195 cm	88 kg		6.966666	42	Arsenal	514	UEFA Eur	2019-202	0	0	0
1438	B. Leno	Bernd	Leno		Goalkeep	29	04/03/19	Bietigheir	Germany	Germany	190 cm	83 kg		7.08	42	Arsenal	524	Premier L	2019-202	0	0	0
1438	B. Leno	Bernd	Leno		Goalkeep	29	04/03/19	Bietigheir	Germany	Germany	190 cm	83 kg			42	Arsenal	957	League Cı	2019-202	0	0	0
1438	B. Leno	Bernd	Leno		Goalkeep	29	04/03/19	Bietigheir	Germany	Germany	190 cm	83 kg			42	Arsenal	1063	FA Cup	2019-202	0	0	0
1438	B. Leno	Bernd	Leno		Goalkeep	29	04/03/19	Bietigheir	Germany	Germany	190 cm	83 kg		7.1	42	Arsenal	514	UEFA Eur	2019-202	0	0	0
1439	Belleru00	Hu00e9ct	Belleru00edn Moru		Defender	26	19/03/19	Barcelon	Spain	Spain	178 cm	74 kg		6.628571	42	Arsenal	524	Premier L	2019-202	1	7	2
1439	Belleru00	Hu00e9ct	Belleru00edn Moru		Defender	26	19/03/19	Barcelon	Spain	Spain	178 cm	74 kg		6.85	42	Arsenal	957	League Cı	2019-202	1	1	0
1439	Belleru00	Hu00e9ct	Belleru00edn Moru		Defender	26	19/03/19	Barcelon	Spain	Spain	178 cm	74 kg		6.5	42	Arsenal	1063	FA Cup	2019-202	1	1	0
1439	Belleru00	Hu00e9ct	Belleru00edn Moru		Defender	26	19/03/19	Barcelon	Spain	Spain	178 cm	74 kg		6.5	42	Arsenal	514	UEFA Eur	2019-202	2	1	0
1117	K. Tierney	Kieran	Tierney		Defender	24	05/06/19	Douglas	Isle of Ma	Scotland	178 cm	70 kg		6.84	42	Arsenal	524	Premier L	2019-202	0	6	2

Figure 18- final data

Cleaning Data

Before working with the gathered Data we need to make sure that the data is cleaned, but first we need to know what data cleaning is. Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

How do we clean data?

- Remove duplicate or irrelevant observations.
- Fix structural errors
- Filter unwanted outliers
- Handle Missing Data

Assess Data

In a perfect world, data would always be complete, accurate, current, pertinent, and unambiguous. In the real world, data is generally flawed on some or all of these dimensions. Data assessment in practice has tended to focus on completeness and

accuracy, and that is the focus of these notes. Currency, pertinence and clarity deserve more attention than they receive, perhaps, but their assessment requires very different methods.

Assessment is sometimes thought of as a preliminary to analysis proper. This is a useful distinction in some circumstances, but in general the assessment of error and the drawing of substantive conclusions are two sides of the same coin. This is suggested by the symbolic equation "Data = Reality + Error", in which "Reality" represents conclusions drawn from the data that are valid despite the error and "Error" represents spurious conclusions suggested by the data as a result of error. Since all conclusions fall into one or the other of these two categories, conclusions about error are at the same time conclusions about reality, and conversely.

There are two general approaches to the assessment of data, direct and indirect. Direct assessment consists of evaluating the coverage and content of a data set. Coverage refers to the faithfulness of the correspondence between the records that constitute the data set and the statistical aggregate the data set represents. Data sets may omit records for some entities that should be represented and include records that should not be included. Improper inclusions occur when a data set includes more than one record for the same entity, includes records for entities not in the statistical aggregate, or includes fictitious records. Content refers to the completeness and accuracy of the information contained on the records in the data set.

Direct assessment

Requires a record-matching study, in which two data sets are compared. The records in each data set are divided into two groups: matched records, which represent entities represented by records in the other data set, and unmatched records, the remainder. Numbers of matched and unmatched records provide a basis for assessing coverage. Comparison of corresponding values on matched records provides a basis for assessing content.

The value of information on coverage provided by a record-matching study is limited by response correlation bias, which exists whenever the inclusion of an entity in one data set is not independent of its inclusion in the other data set. In the extreme case of perfect correlation, both data sets would consist entirely of records representing the same entities, and matching would provide no information on coverage.

Strict independence is unattainable in practice, but a modicum of independence is necessary for a record-matching study to yield useful information on coverage.

Record-matching studies for population censuses require a post-enumeration survey, a survey taken after the census for the purpose of evaluating its quality. Matching studies for civil registration data may involve special surveys or draw on other sources of data.

Record-matching studies may be used to assess content error in population surveys. Coverage is less important for survey data than for population census or civil registration data because information is obtained only for a sample.

Indirect Assessment

Direct assessment of data sets is expensive, both because a second data set is required for comparison and because matching is often a complex and difficult process. The results of direct assessment are, moreover, limited by response correlation bias and by the tendency of data sets collected at the same or nearly the same time to have similar content error. The indirect approach, by which data sets are assessed by analyzing the accuracy of statistics derived from them, is generally far less expensive and will often give results as good as or better than direct assessment.

Assessment of a statistic is concerned with its accuracy, that is, with how close it is to the true value it represents. The principle means for assessing the accuracy of a statistic is comparison with other statistics. Comparison may take many forms. In some cases it may rely on general knowledge rather than on specific comparison statistics. FIGURE 1

Direct comparisons with other statistics generally provide the strongest conclusions about data quality.

Statistics derived from higher quality data sets will generally be more accurate than statistics derived from lower quality data sets, but there is no

simple, general relation between the quality of a data set and the accuracy of statistics derived from it.

Assess the Quality of Data

There are 6 Metrics to Assess the Quality of Data:

1- Ratio of Data to Errors

2- Number of empty values

3- Data Transformation Error Rates

4- Amounts of Dark Data

5- Email Bounce Rates

6- Data Storage

Rating Model

After doing the analysis on our data and cleaning it, now we can work with our data to make a rating model that rates the player based on his statistics in the field and to achieve that we are going to use Gradient Boosting Classifier machine learning model.

Weight?	Feature
+6.852	<BIAS>
+0.244	games/minutes_played
+0.231	dribbles/success
+0.230	passes/key
+0.157	shots/total
+0.152	duels/won
+0.149	goals/total
+0.046	tackles/total
+0.045	tackles/blocks
+0.017	goals/assists
+0.015	substitutes/in
+0.013	cards/yellowred
… *4 more positive* …	
… *5 more negative* …	
-0.021	tackles/interceptions
-0.045	fouls/committed
-0.047	passes/total
-0.117	shots/on
-0.138	games/appearences
-0.159	games/lineups
-0.204	duels/total
-0.256	dribbles/attempts

```
[ ] # Building the optimal model using Backwa
```

Figure 19- Rating model 1

```
[ ] # Create linear regression object
    lin_reg_mod = linear_model.LinearRegression()
    lin_reg_mod.fit(X_train, y_train)

    LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

[ ] y_pred = lin_reg_mod.predict(X_test)
    #filtered_by_league_NULL_FW['New_Rate']=lin_reg_mod.predict(X)
```

Linear Regression in Tensorflow

Figure 20- Linear regression

```
[ ] test_set_rmse = (np.sqrt(mean_squared_error(y_test, y_pred)))

    test_set_r2 = r2_score(y_test, y_pred)

[ ] print('The Error percentage(RMSE) in our model = {}'.format(test_set_rmse))
    #print(test_set_r2)

    The Error percentage(RMSE) in our model = 0.18487695014819827
```

Figure 21-Rating model 2

```
[ ] import eli5
    eli5.explain_weights(lin_reg_mod, feature_names=list(X.columns))

    /usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:144: FutureWarning: The
      warnings.warn(message, FutureWarning)
    y top features
```

Figure 22- Rating model 3

Evaluation Of The Model

We'll now check the predictions against the actual values by using the RMSE and R-2 metrics, two metrics commonly used to evaluate regression tasks:

```
[ ] Result = pd.DataFrame({'Actual Rate':y_test ,'Predicted_Rate':y_pred})
    Result
```

Figure 23- Rating model 4

[ ]

	Actual Rate	Predicted_Rate
1881	6.658333	6.660760
879	6.655555	6.663060
1085	6.742857	6.643975
1612	6.725000	6.697518
1470	7.041666	7.329397
2101	6.861111	6.599849
2044	6.800000	7.007221
1103	6.525000	6.657147
617	6.872727	6.777634
720	6.771875	6.573388
1250	7.715789	7.349490
1956	6.972222	7.077650
80	6.796428	6.804159
2072	6.748275	6.638158
2152	6.621052	6.646217
1062	7.588888	7.975552
2026	6.900000	6.716971
2010	6.666666	6.671417
1836	6.755555	6.697875
739	6.650000	6.706760
1536	6.300000	6.685294
338	6.800000	6.795656

Figure 24- actual vs predicted

Prediction Model

In order to make our prediction model we are going to use some tools **numpy**, **pandas**,**matplotlib** and **sklearn** to make a multiple linear regression model that predicts the player rating which is overall his performance in the next year, so we have done the following steps.

- Import needed libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import metrics, svm
from sklearn import preprocessing
from sklearn import utils
import xlrd
from xlrd import open_workbook
```

Figure 25- import

- Read players data and filter it by choosing Attackers with a rating that is above zero

```
dataset = pd.read_csv('all_Data.csv')
```

```
Data_Filter_2 = dataset[dataset.position=='Attacker']
Data_Filter_2 = Data_Filter_2[Data_Filter_2.rating > 0]
Data_Filter_2.rating
```

```
33      7.715789
34      7.000000
35      7.600000
36      8.700000
76      6.843750
          ...
2224    7.600000
2225    6.450000
2239    6.766666
2243    6.806250
2244    6.300000
Name: rating, Length: 339, dtype: float64
```

Figure 26- Read Player Data

- Eliminate unwanted attributes that won't affect the player rating that is our X parameter in the regression model.

```
x = x.drop(columns=['birth_date','birth_place','birth_country', 'nationality', 'height',
                    'injured', 'rating', 'team_id', 'team_name', 'league_id', 'league', 'season', 'captain',
                    'goals/conceded','goals/saves', 'cards/yellowred', 'penalty/commited', 'penalty/saved','weight'])
```

Figure 27- features optimization

- The total number of parameters in X now is 26 parameters that have the most effect on rating the player performance.

```
x.columns
```

```
Index(['age', 'shots/total', 'shots/on', 'goals/total', 'goals/assists',
       'passes/total', 'passes/key', 'passes/accuracy', 'tackles/total',
       'tackles/blocks', 'tackles/interceptions', 'duels/total', 'duels/won',
       'dribbles/attempts', 'dribbles/success', 'fouls/drawn',
       'fouls/committed', 'cards/yellow', 'cards/red', 'penalty/won',
       'penalty/success', 'penalty/missed', 'games/appearences',
       'games/minutes_played', 'games/lineups', 'substitutes/in'],
      dtype='object')
```

Figure 28- final features

- Take the rating column as our Y parameter in the regression model.

```
y = Data_Filter_2['rating']
y
```

```
33      7.715789
34      7.000000
35      7.600000
36      8.700000
76      6.843750
          ...
2224    7.600000
2225    6.450000
2239    6.766666
2243    6.806250
2244    6.300000
Name: rating, Length: 339, dtype: float64
```

Figure 29- Rating

- Then we Train our model using our X and Y parameters using 20% of the data as a test data and 80% as training data.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 0)
```

Figure 30- Train

- Then we use Scalar to scale the very high scores in data so that the model cannot remove it to read this value as outlier and eliminate it from his calculations.

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

Figure 31- Scaling

- Then we create our regression model and predict the player rating in the player rating.

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
LinearRegression()
```

```
y_pred = regressor.predict(X_test)
```

Figure 32- Predict

- As we can see here, the prediction of the player rating compared with the actual rating that takes different parameters to rate the player performance.

```
Result = pd.DataFrame({'Actual Rate':y_test ,'Predicted_Rate':y_pred})
Result
```

	Actual Rate	Predicted_Rate
475	7.050000	6.970055
1762	6.650000	6.879814
1472	7.075000	7.014342
1066	6.500000	6.761418
1563	6.900000	6.934651
...	...	...
884	6.533333	6.911777
405	7.600000	7.072073
722	7.800000	6.967387
406	7.600000	7.011086
1798	7.100000	7.216424

Figure 33- Actual Vs Predicted

- Then we check the accuracy of our model using r squared and r squared adjacent.

```
import statsmodels.api as sm
X1 = sm.add_constant(X_train)
result = sm.OLS(y_train, X1).fit()
result.rsquared, result.rsquared_adj
```

```
(0.3208579100263247, 0.2515576967637049)
```

Figure 34- Accuracy

- Then we create a prediction function that takes the id of the player and returns the prediction of his rating (performance) in the next year. The function takes the id of the player and then searches in the file of the gathered data and takes the parameters of the player and sends it to our prediction model to return the result.

```
def Prediction_function(test_id) :
    book = open_workbook('all_Data.xlsx')
    sheet = book.sheet_by_index(0)
    id_Column_number = 2
    for row in range(sheet.nrows):
         if sheet.cell(row,id_Column_number).value == test_id:
            row_number = row
            break
    df = pd.read_excel('all_Data.xlsx', sheet_name='Sheet1')
    row_to_be_predicted = df.iloc[row_number]
    row_to_be_predicted.to_frame()
    df2 = row_to_be_predicted[['age', 'shots/total', 'shots/on', 'goals/total', 'goals/assists',
       'passes/total', 'passes/key', 'passes/accuracy', 'tackles/total',
       'tackles/blocks', 'tackles/interceptions', 'duels/total', 'duels/won',
       'dribbles/attempts', 'dribbles/success', 'fouls/drawn',
       'fouls/committed', 'cards/yellow', 'cards/red', 'penalty/won',
       'penalty/success', 'penalty/missed', 'games/appearences',
       'games/minutes_played', 'games/lineups', 'substitutes/in']]
    a = np.array(df2.values.tolist())
    b= np.array(a).reshape(-1, 26)
    b = sc_X.transform(b)
    PREDICTION = regressor.predict(b)
    return PREDICTION
```

Figure 35- Prediction Model

- Here we test our model by sending ID number 1467 to the model and get the prediction of the player with this specific ID.

```
RESULT = Prediction_function(1467)
RESULT
```

```
array([6.82680427])
```

Figure 36- Prediction for ID

Mobile Application

Technical Specification

Overview

To represent our project output after applying our rating model and prediction model with a mobile application developed with **Flutter**.

Flutter is a free and open-source mobile UI framework created by Google and released in May 2017. In a few words, it allows you to create a native mobile application with only one codebase. This means that you can use one programming language and one codebase to create two different apps (for iOS and Android).

Flutter is considered to be the best language to develop mobile applications as it can be executed in both software's **Android** and **IOS** and that is a huge advantage for **Flutter** as we save a lot of time coding the same application for android with **android studio** and coding with **swift**.

So our application can be used in many devices that use **android** or **IOS**.

Flutter consists of two important parts:

- An SDK (Software Development Kit): A collection of tools that are going to help you develop your applications. This includes tools to compile your code into native machine code (code for iOS and Android).

- A Framework (UI Library based on widgets): A collection of reusable UI elements (buttons, text inputs, sliders, and so on) that you can personalize for your own needs.

To develop with Flutter, you will use a programming language called Dart. The language was created by Google in October 2011, but it has improved a lot over these past years.

Dart focuses on front-end development, and you can use it to create mobile and web applications.

Installation

We use in this project Visual studio 1.58, Flutter 2.2.3.

Minimum SDK version: 16.

Target SDK version: 29.

Packages

```yaml
name: player_performance_predictor
version: 1.0.0+1

environment:
  sdk: ">=2.7.0 <3.0.0"

dependencies:
  flutter:
    sdk: flutter
  cupertino_icons: ^1.0.0
  http: ^0.13.3
  provider: ^4.3.3
  splashscreen: ^1.3.5

uses-material-design: true


  assets:
    - assets/pics/
    - assets/pics/leagues/

  fonts:
    - family: Raleway
      fonts:
        - asset: assets/fonts/Raleway-Regular.ttf
        - asset: assets/fonts/Raleway-Bold.ttf
          weight: 700
        - asset: assets/fonts/Raleway-Black.ttf
          weight: 900
    - family: RobotoCondensed
      fonts:
        - asset: assets/fonts/RobotoCondensed-Regular.ttf
        - asset: assets/fonts/RobotoCondensed-Bold.ttf
          weight: 700
        - asset: assets/fonts/RobotoCondensed-Light.ttf
          weight: 300


}
```

Figure 37- App packages

Getting started Database

Rapid API-Football

We just choose the suitable API according to our needs.

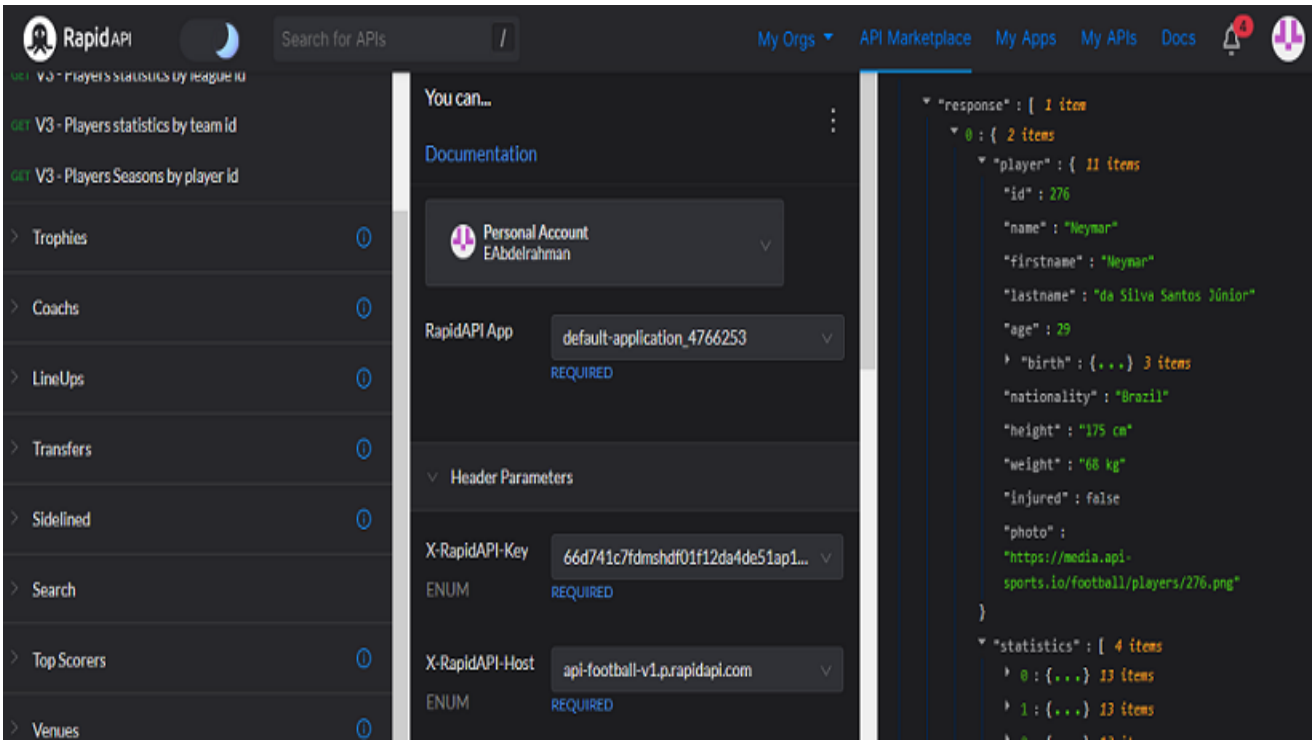


Figure 38- APP with API

Connection to Rapid API-Football

We just call end points and send specific request we need to get the response and export it the widgets.

```
Future<void> fetchTeams(int leagueId) async {
    var url =
        'https://api-football-v1.p.rapidapi.com/v2/teams/league/$leagueId';

    Map<String, String> requestHeaders = {
      'Content-type': 'application/json',
      'x-rapidapi-host': ' api-football-v1.p.rapidapi.com',
      'x-rapidapi-key': '66d741c7fdmshdf01f12da4de51ap16b6adjsn3c3d982c712d',
    };
    try {
      http.Response response = await http.get(url, headers: requestHeaders);
      final extractedData = json.decode(response.body) as Map<String, dynamic>;
      var teamData = extractedData['api']['teams'] as List;
      if (teamData == null) {
        return;
      }
      final List<Team> loadedTeams = [];
      teamData.forEach((team) {
        loadedTeams.add(
          Team(
              teamId: team['team_id'],
              teamLogo: team['logo'],
              teamName: team['name']),
        );
      });
      _teamItems = loadedTeams;
      notifyListeners();
    } catch (e) {
      throw (e);
    }
  }
```

```dart
Future<void> fetchplayerStatistics(int playerid, int season) async {
   var url =
       'https://api-football-v1.p.rapidapi.com/v3/players?id=$playerid&season=$season';

   Map<String, String> requestHeaders = {
     'Content-type': 'application/json',
     'x-rapidapi-host': ' api-football-v1.p.rapidapi.com',
     'x-rapidapi-key': '66d741c7fdmshdf01f12da4de51ap16b6adjsn3c3d982c712d',
   };
   try {
     http.Response response = await http.get(url, headers: requestHeaders);
     final extractedData = json.decode(response.body) as Map<String, dynamic>;
     var responseData = extractedData['response'] as List;
     if (responseData == null) {
       return;
     }

     final List<Player> loadedPlayers = [];
      responseData.forEach((player) {
       var playerStates = player['statistics'] as List;

       loadedPlayers.add(
         Player(
           playerId: player['player']['id'],
           playerImage: player['player']['photo'],
           playerName: player['player']['name'],
           teamLogo: playerStates[0]['team']['logo'],
           nationality: player['player']['nationality'],
           age: player['player']['age'],
           position: playerStates[0]['games']['position'],
           rate: playerStates[0]['games']['rating'],
           gamesAppearences: playerStates[0]['games']['appearences'],
           gamesMinutes: playerStates[0]['games']['minutes'],
           totalShots: playerStates[0]['shots']['total'],
           shotsOnTarget: playerStates[0]['shots']['on'],
           totalGoals: playerStates[0]['goals']['total'],
           goalsAssists: playerStates[0]['goals']['assists'],
           goalsSaves: playerStates[0]['goals']['saves'],
           goalsconceded: playerStates[0]['goals']['conceded'],
           totalPasses: playerStates[0]['passes']['total'],
           keyPasses: playerStates[0]['passes']['key'],
           passesAccuracy: playerStates[0]['passes']['accuracy'],
           totalTackles: playerStates[0]['tackles']['total'],
           blockTackles: playerStates[0]['tackles']['blocks'],
           interceptions: playerStates[0]['tackles']['interceptions'],
           totalDuels: playerStates[0]['duels']['total'],
           duelsWon: playerStates[0]['duels']['won'],
           dribblesSuccess: playerStates[0]['dribbles']['success'],
           foulsDrawn: playerStates[0]['fouls']['drawn'],
           foulsCommitted: playerStates[0]['fouls']['committed'],
           yellowCards: playerStates[0]['cards']['yellow'],
           redCards: playerStates[0]['cards']['red'],
         ),
       );
     });
     _playerItems = loadedPlayers;
     notifyListeners();
      } catch (e) {
     throw (e);
   }
```

```dart
@override
 void initState() {
   setState(() {
     _isLoading = true;
   });

   Future.delayed(Duration.zero, () {
     setState(() {
       playerid = ModalRoute.of(context).settings.arguments as int;
     });

     Provider.of<Players>(context, listen: false)
         .fetchplayerStatistics(playerid, season)
         .then((_) {
       player = Provider.of<Players>(
         context,
         listen: false,
       ).findById(playerid);
       setState(() {
         _isLoading = false;
       });
     });
   });
   super.initState();
 }
 @override
 Widget build(BuildContext context) {
   return _isLoading
       ? Center(
           child: CircularProgressIndicator(),
         )
       : Scaffold(
           appBar: AppBar(
             title: Text('Statistics',
                 style: Theme.of(context).textTheme.bodyText1),
             backgroundColor: Color(0xff1C1C1E),
             actions: [
               PopupMenuButton(
                 itemBuilder: (BuildContext ctx) => [
                   PopupMenuItem(
                     child: Text(
                       'Season 2020',
                     ),
                     value: 2020,
                   ),
                   PopupMenuItem(
                     child: Text(
                       'Season 2019',
                     ),
                     value: 2019,
                   ),

                 onSelected: (itemIdentifier) {
                   if (itemIdentifier == 2020) {
                     setState(() {
                       season = 2020;
                       _isLoading = true;

                       Provider.of<Players>(context, listen: false)
                           .fetchplayerStatistics(playerid, season)
                           .then((_) {
                         player = Provider.of<Players>(
                           context,
                           listen: false,
                         ).findById(playerid);
                         setState(() {
                           _isLoading = false;
                         });
                       });
                     });
                   } else if (itemIdentifier == 2019) {
                     setState(() {
                       season = 2019;

                       _isLoading = true;

                       Provider.of<Players>(context, listen: false)
                           .fetchplayerStatistics(playerid, season)
                           .then((_) {
                         player = Provider.of<Players>(
                           context,
                           listen: false,
                         ).findById(playerid);
                         setState(() {
                           _isLoading = false;
                         });
                       });
                     });
                   }
                 },
               ),
             ],
           ),
           body: _isLoading
               ? Center(
                   child: CircularProgressIndicator(),
                 )
               : Container(
                   color: Color(0xff1C1C1E),
                   height: MediaQuery.of(context).size.height,
                   child: PlayerStatsItem()
```

```dart
Future<List> searchforPlayers(String query) async {
  var url = 'https://api-football-v1.p.rapidapi.com/v2/players/search/$query';

  Map<String, String> requestHeaders = {
    'Content-type': 'application/json',
    'x-rapidapi-host': ' api-football-v1.p.rapidapi.com',
    'x-rapidapi-key': '66d741c7fdmshdf01f12da4de51ap16b6adjsn3c3d982c712d',
  };
  try {
    http.Response response = await http.get(url, headers: requestHeaders);
    final extractedData = json.decode(response.body) as Map<String, dynamic>;
    var responseData = extractedData['api']['players'] as List;
    if (responseData == null) {
      return [];
    }
    return responseData;
  } catch (e) {
    throw (e);
  }
}
```

Figure 42- Connection to API

```dart
class SearchScreen extends StatefulWidget {

    final playerCard = Provider.of<Players>(context);
    return Scaffold(
      backgroundColor: Colors.black,
      resizeToAvoidBottomInset: false,
      appBar: AppBar(
        title: Text(
          'Search',
          style: Theme.of(context).textTheme.bodyText1,
        ),
        centerTitle: true,
        backgroundColor: Theme.of(context).primaryColor,
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.start,
          children: <Widget>[
            SearchForm(
              onSearch: (q) {
                setState(() {
                  query = q;
                });
              },
            ),
            query == null
                ? Expanded(
                    child: Column(
                      mainAxisAlignment: MainAxisAlignment.center,
                      children: [
                        Icon(
                          Icons.search,
                          color: Colors.white,
                          size: 110,
                        ),
                        Text(
                          'No results to display',
                          style: TextStyle(
                            color: Colors.white,
                            fontSize: 20,
                            fontWeight: FontWeight.bold,
                          ),
                        )
                      ],
                    ),
                  )
                : FutureBuilder(
                    future: playerCard.searchforPlayers(query),
                    builder: (context, snapshot) {
                      if (snapshot.connectionState == ConnectionState.waiting) {
                        return Center(
                          child: CircularProgressIndicator(),
                        );
                      }

                      if (snapshot.hasData) {
                        return Expanded(
                          flex: 2,
                          child: ListView.builder(
                              itemCount: snapshot.data.length,
                              itemBuilder: (context, i) {
                                return SearchResult(snapshot.data[i]);
                              }),
                        );
                      }

}
```

Application User Journey

Home page

When client starts the application, this splash screen will appear for 5 seconds while initializing the application.

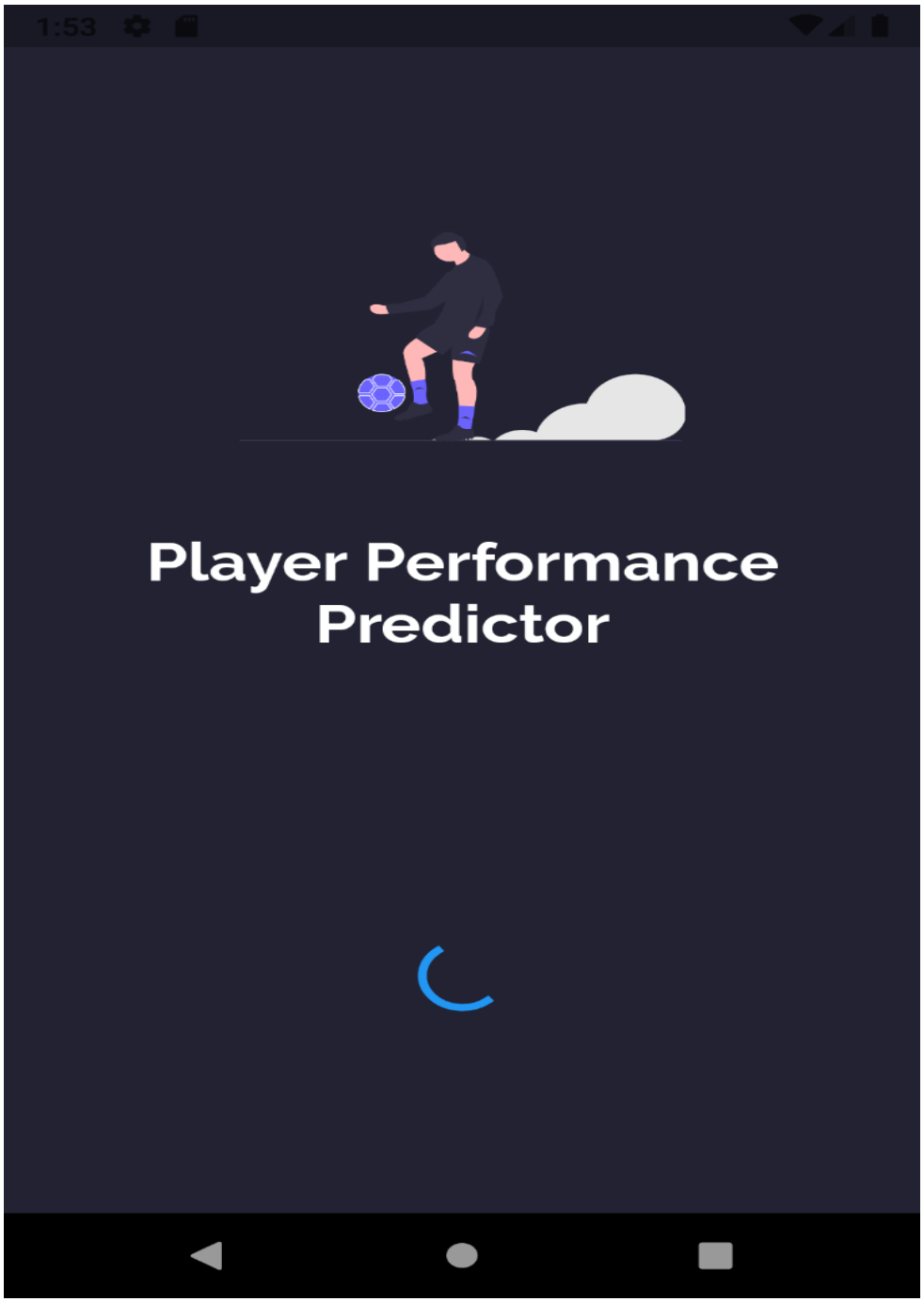


Figure 44- Home Page

Leagues page:

After starting the application, the home screen will appear to the user with six different leagues.

The user has two options: either reach the player by choosing the league then the team then the team squad and finally the player statistics, or search by player name.

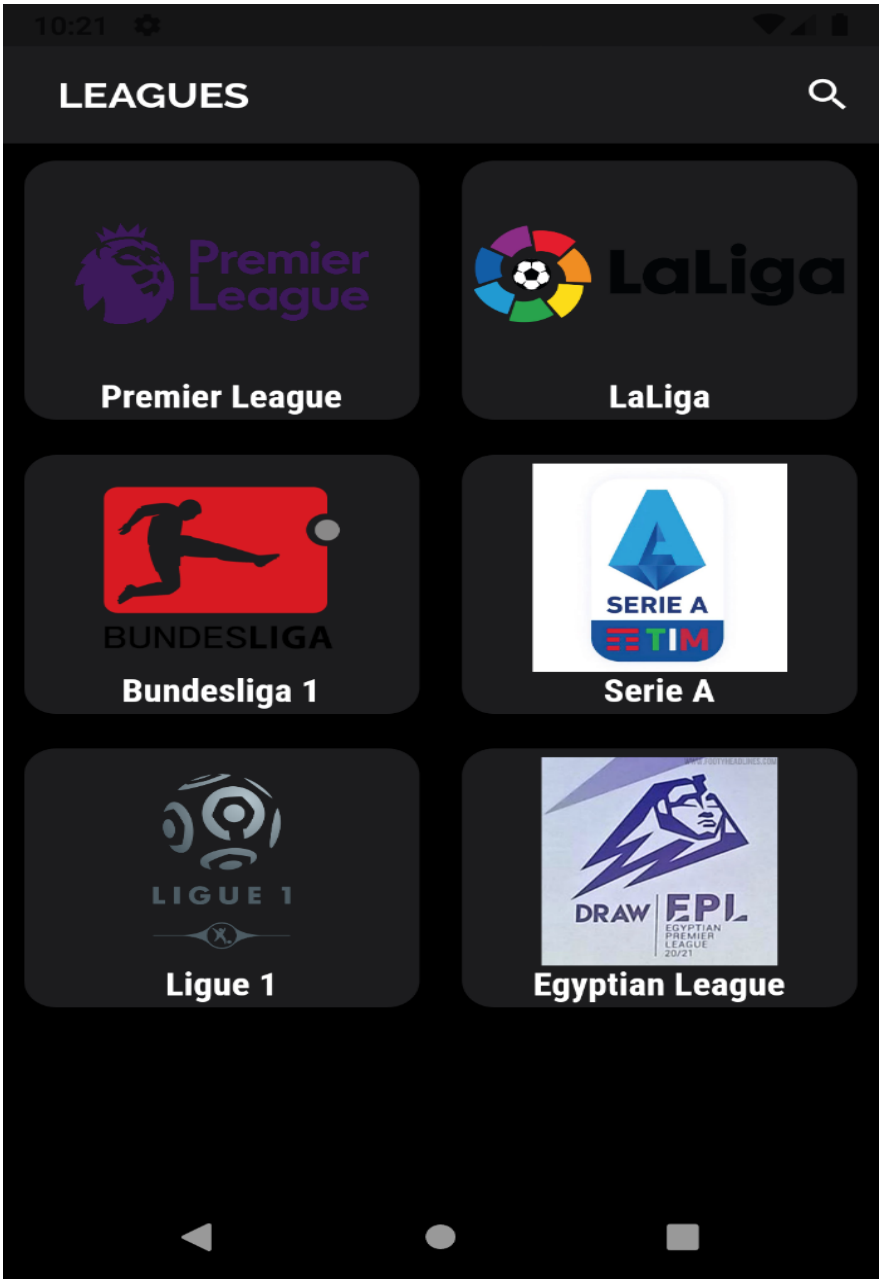


Figure 45- League Page

The first option goes as follows:

Team's page:

User choose Egyptian league for example, the teams in season 2020 will appear:

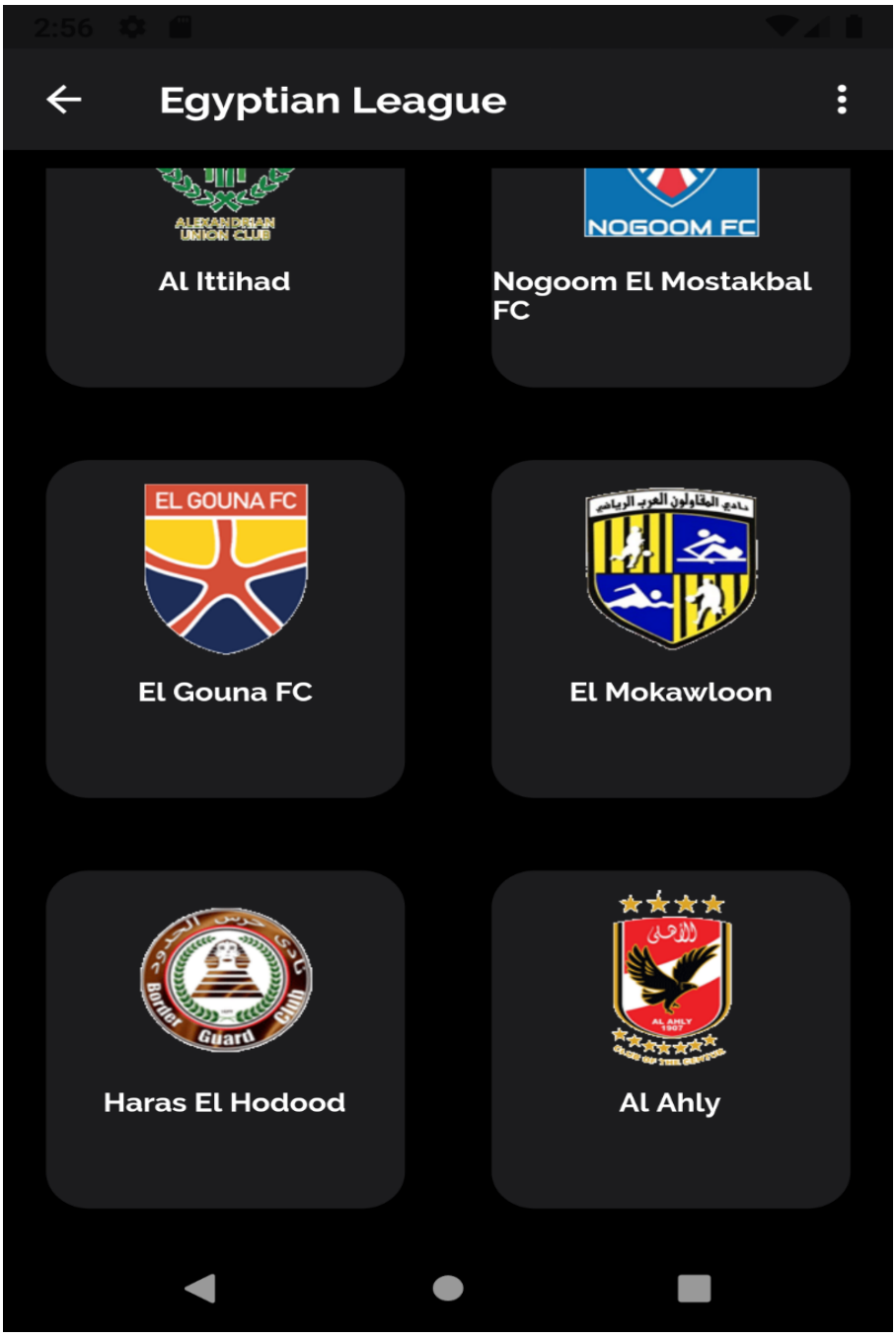


Figure 46-Teams

And if he chooses a different season, 2016 for example, teams in 2016 will appear.

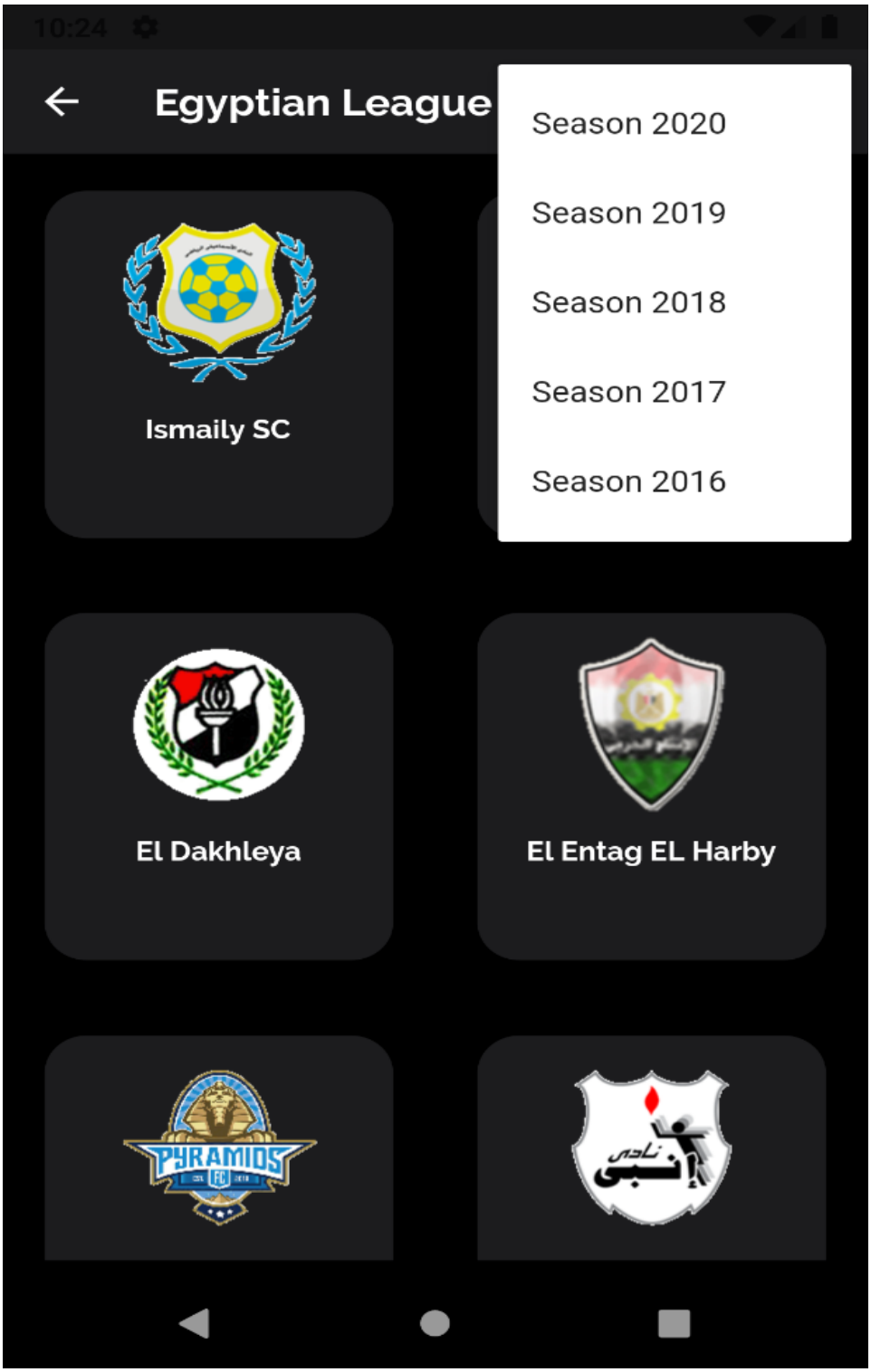


Figure 47- Teams

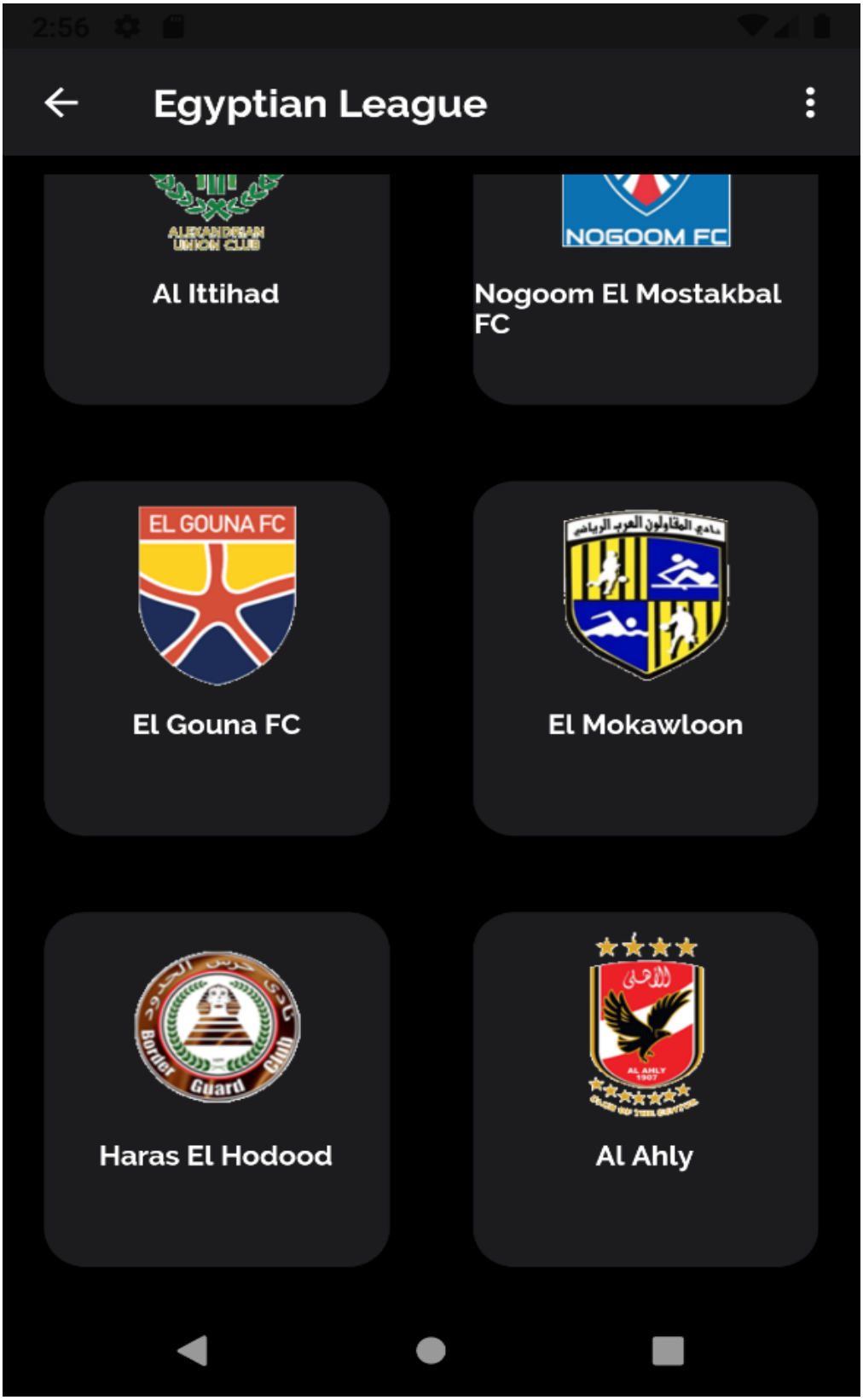


Figure 48- Teams

Squad page:

If the user chooses Al Ahly for example, the players of the team will appear. Each player has a card with his information.

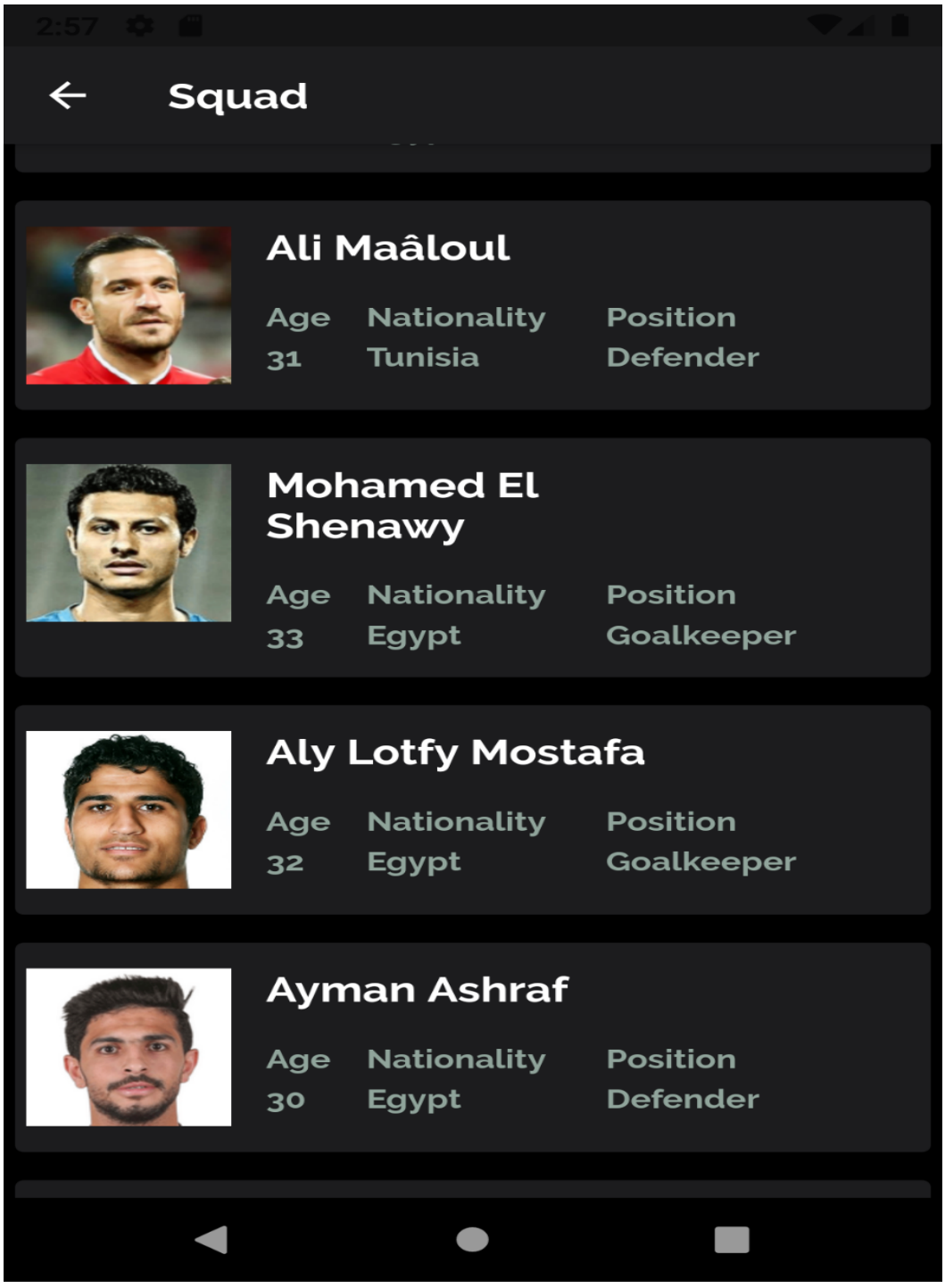


Figure 49- Squad

Stats page:

User chooses a player he wants; the statistics of him will appear.

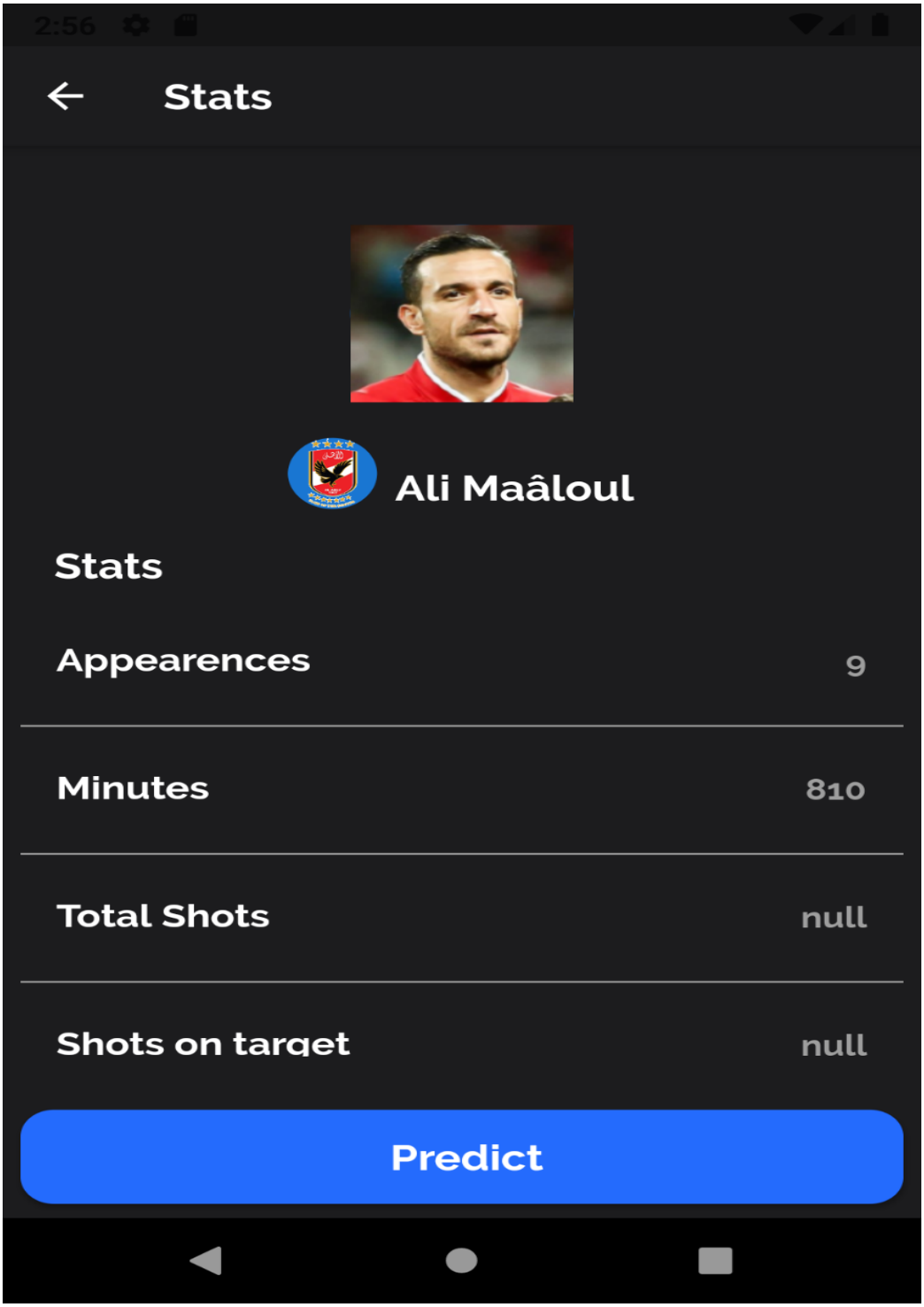


Figure 50-Stats

And if he chooses a different season, 2018 for example, his statistics in 2018 will appear.

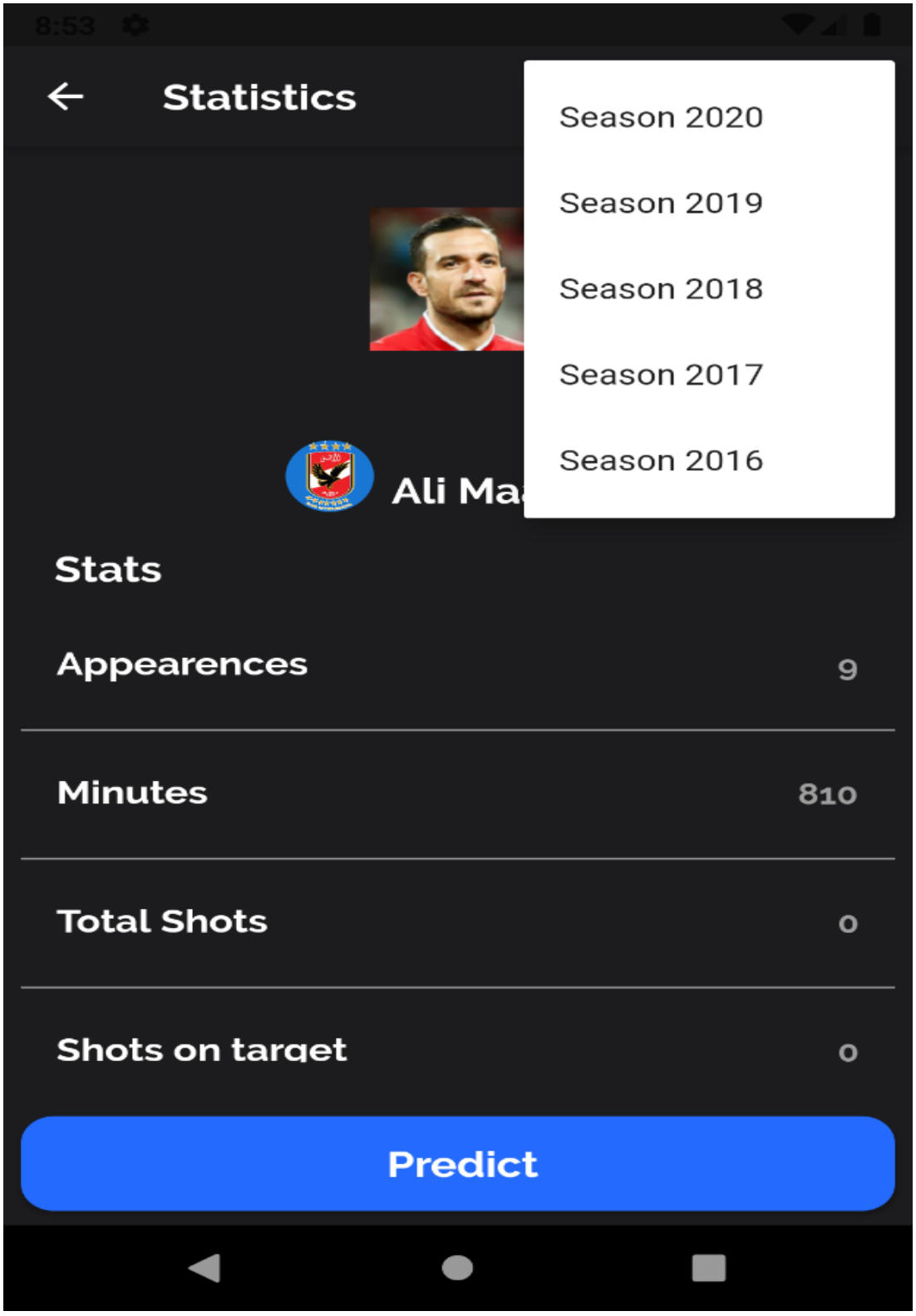


Figure 51- Predict

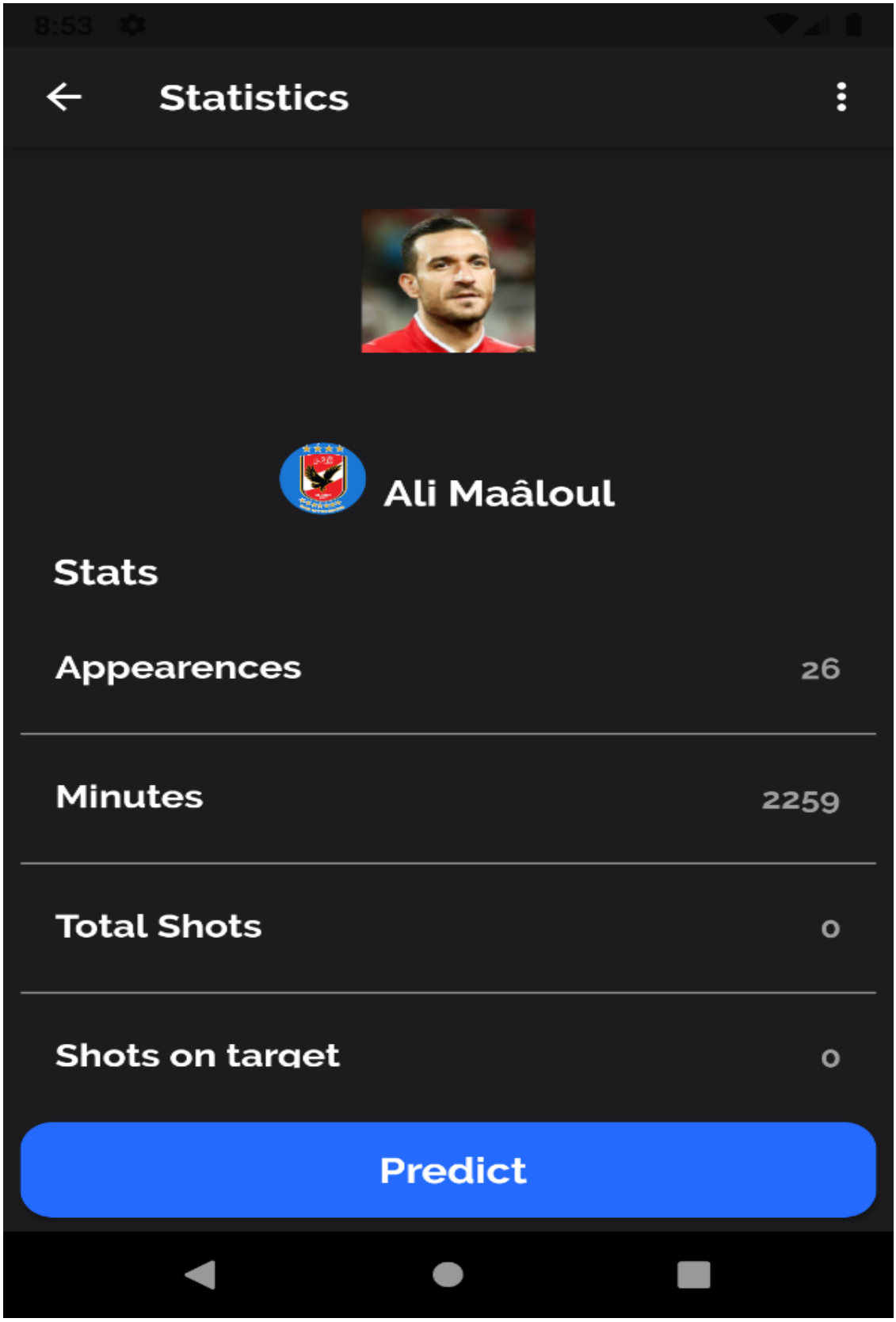


Figure 52- Stats

Search page:

User clicks the search button, the search screen will show up.

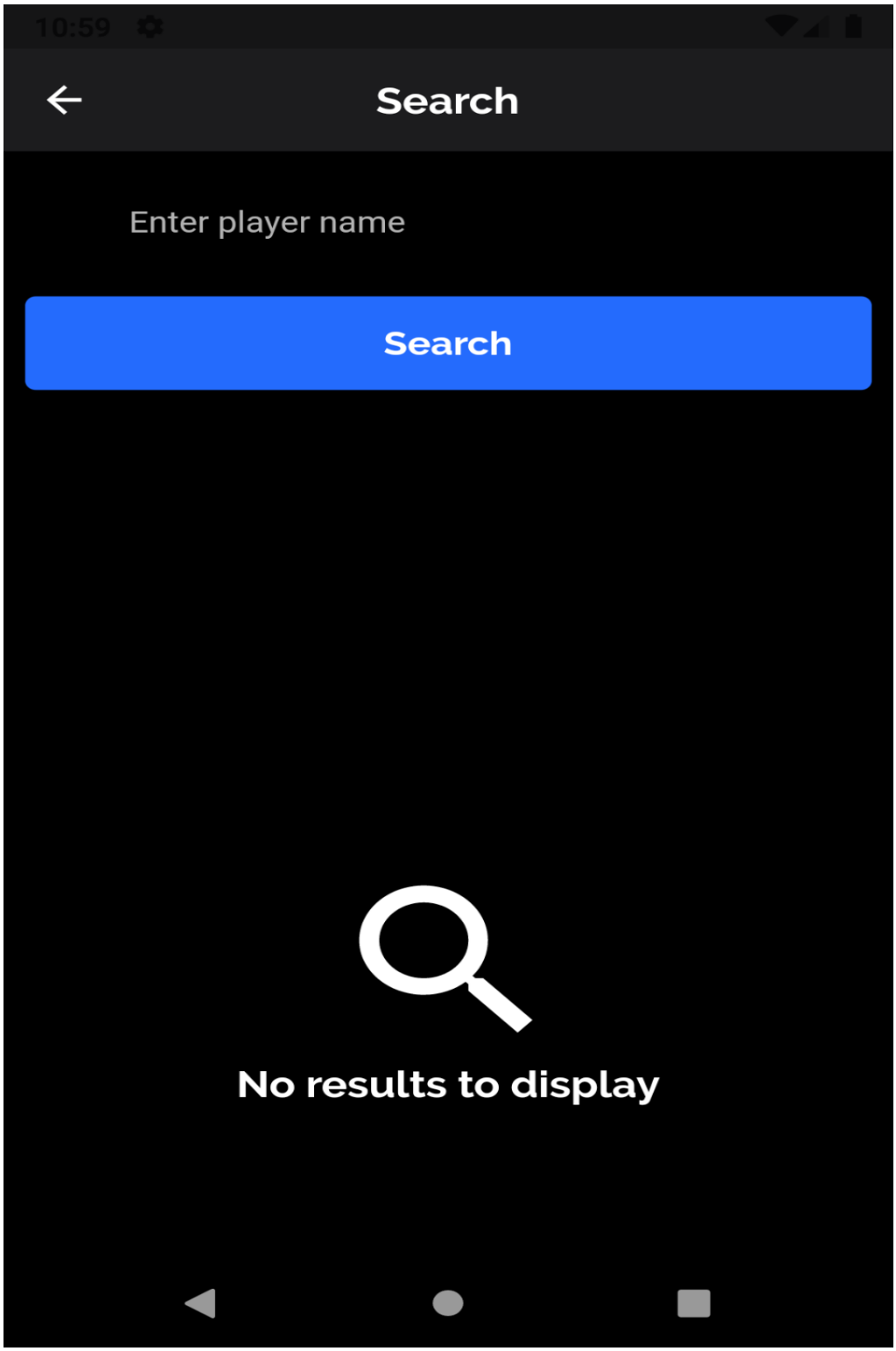


Figure 53- Search

User writes the player name (which should not be less than 4 characters.

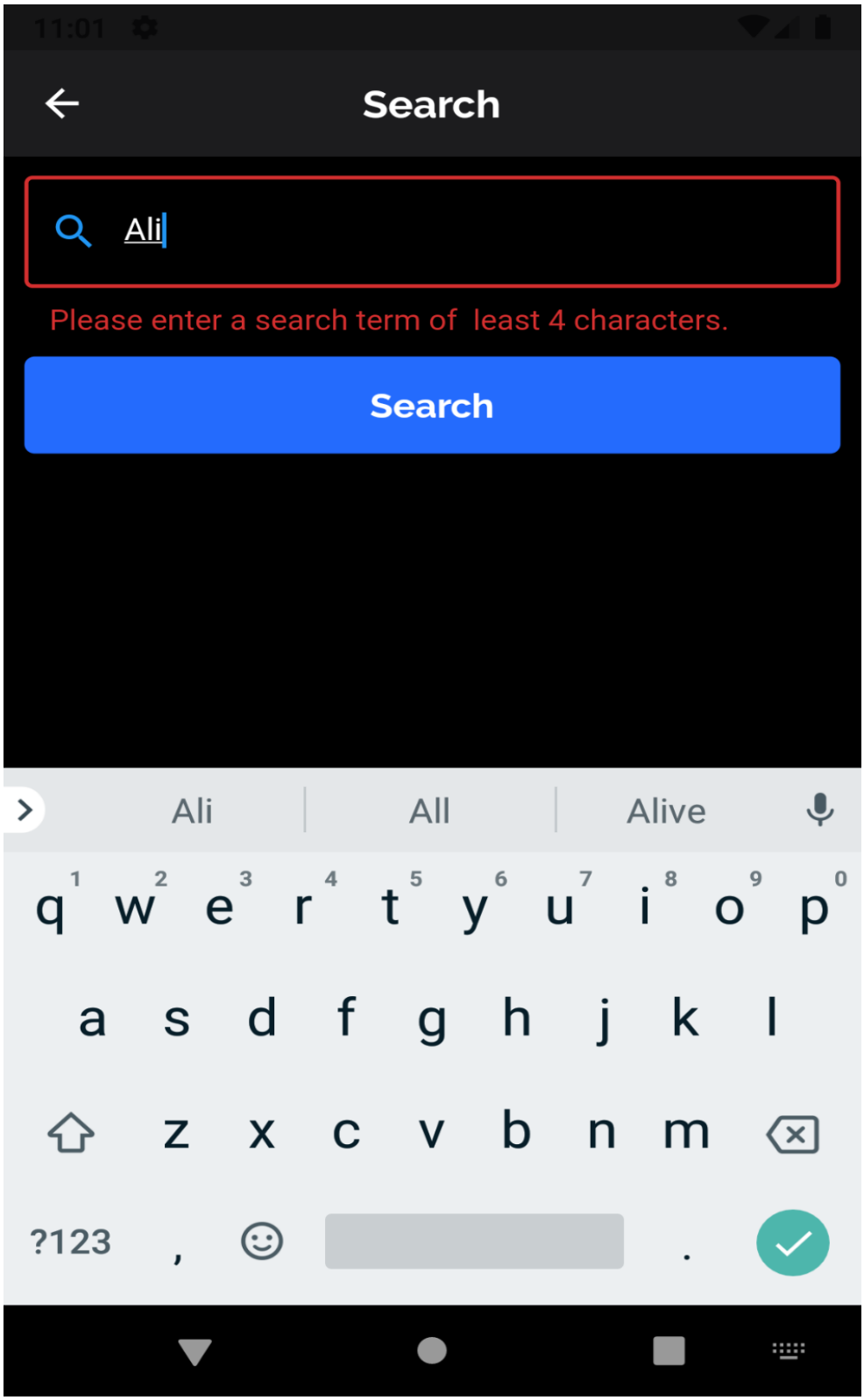


Figure 54-Search

After writing the player name, user click search button, the result will show up.

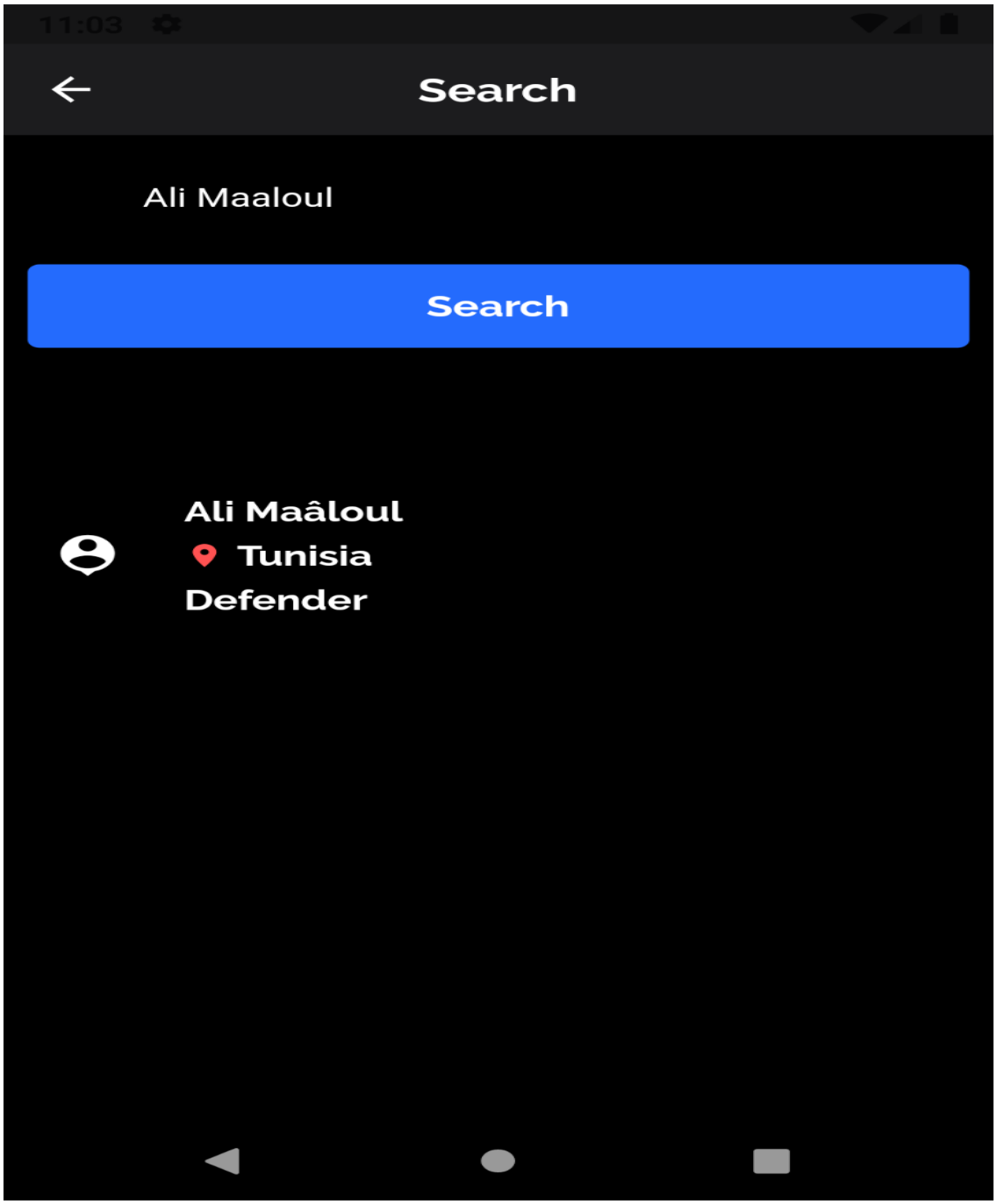


Figure 55-Search

User clicks the player card which will deliver him to the player stats screen.

Video Analysis

For football matches Video Analysis we are going to use several tools: **Python** and its libraries **Open CV, numpy** and **tensor flow**. And we prepared these tools to use **YOLO** real time object detection.

What is **YOLO**?

YOLO is a real time object detection system. On a Pascal Titan X it processes images at 30 FPS and has a map of 57.9% on COCO test-dev.

YOLOv3 is extremely fast and accurate. In map measured at .5 IOU YOLOv3 is on par with Focal Loss but about 4x faster. Moreover, you can easily tradeoff between speed and accuracy simply by changing the size of the model, no retraining required!

How does it work?

Prior detection systems repurpose classifiers or localizers to perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections.

We use a totally different approach. We apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

This model has several advantages over classifier-based systems. It looks at the whole image at test time so its predictions are informed by global context in the image. It also makes predictions with a single network evaluation unlike systems like R-CNN which require thousands for a single image. This makes it extremely fast, more than 1000x faster than R-CNN and 100x faster than Fast R-CNN.

Our Code works on each frame of the video and detects Home team players, Away team players and referees as shown in the figure below. The model is trained using 3 layered convolutional neural networks with 500 epochs with samples of each team and referee and the ball shape in the video.

Figure 56- Video detection

After Detecting the Players and the ball we track the player in the field and represent his location in the following image by putting a colored circle that shows the location of the player in the field.

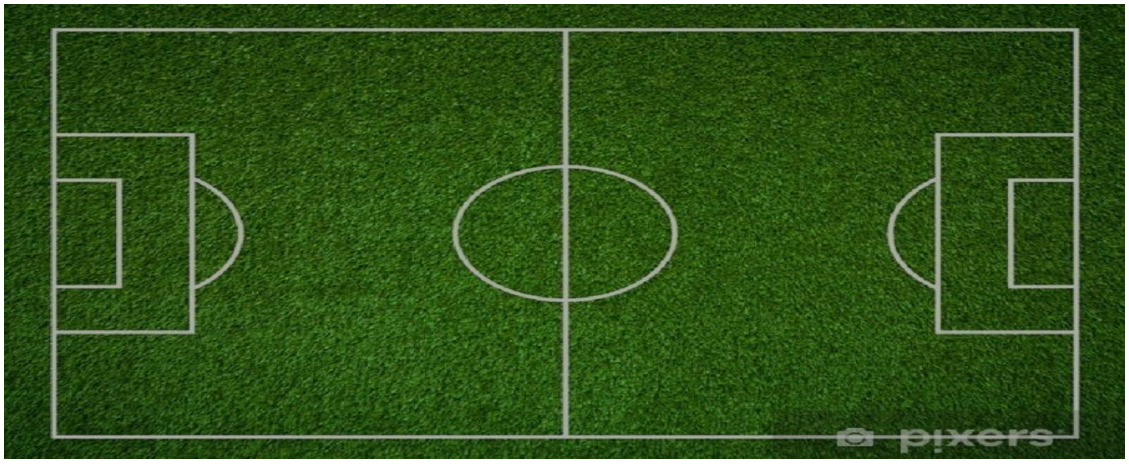


Figure 57-Video detection

We Track the player by giving the coordinates of the middle circle of the field and the top and bottom end of the field from the middle exactly from the video and project it to the above image and there is a function that calculates the distances from these points to track the player location and show it as a colored circle as the example below.

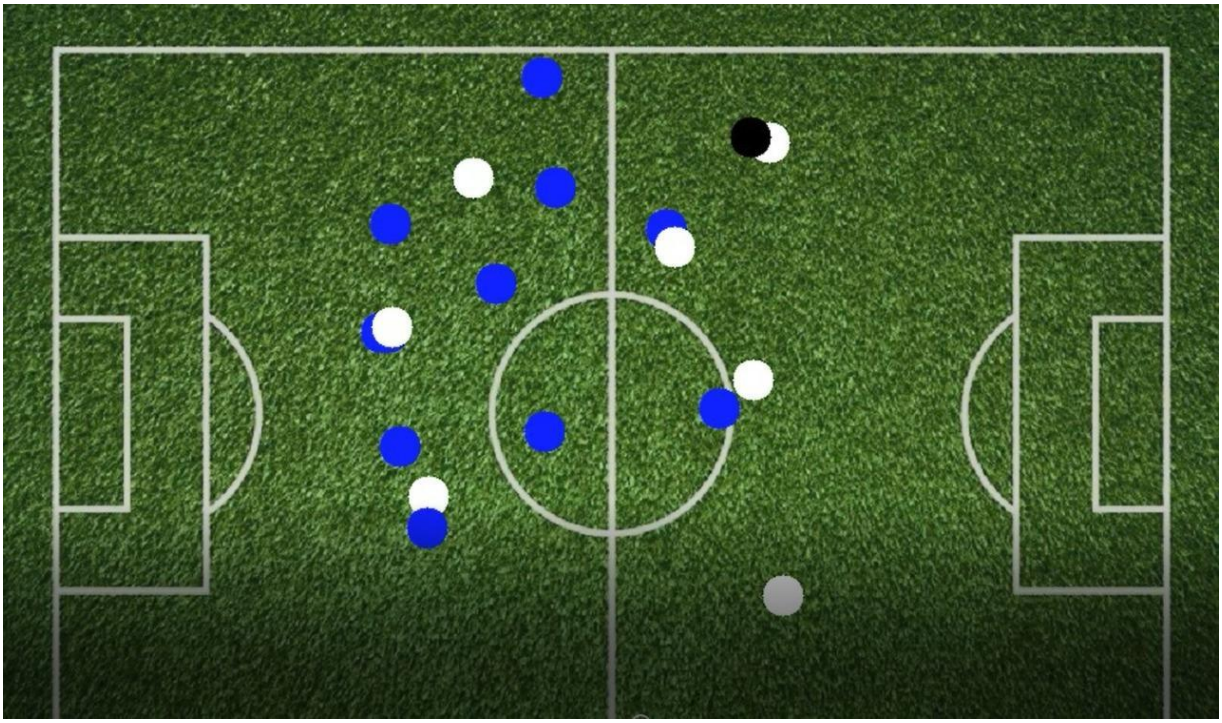


Figure 58- Video detection

Video Analysis Model

In the following section we are going to explain the used models to detect the following objects: players of first team, second team, and referee and the ball and make a live graph that shows the positions of the objects.

We have 3 main sections in the code:

- Detection Model:
  That detects the objects from the image which is **YOLO** that is used to detect several objects. This model detects the humans and the ball in each frame in the video.

- Classification Model:
  That Classifies the humans into 3 classes: Team [1] , Team [2] , Referee.
  The model is trained by a data set from players of each team of the 2 teams and referee and the ball

- A Function that gets perspective from the video and draws it in the template photo shown in **Figure 45**.

The following steps are made to reach the final output:

- Import the needed Libraries

```
import cv2
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import load_model
from numpy import argmax
```

Figure 59- Libraries Import

- Load the classification model which is named **model.h5** and read the video that needs to be analyzed and the image in **Figure 45**.

```
model = load_model(r'files\model.h5')

cap = cv2.VideoCapture(r'data\video.avi')
ground=cv2.imread(r'data\dst.jpg')
```

Figure 60- Classification model

- Then load YOLO detection model

```
# Load Yolo
net = cv2.dnn.readNet("files\yolov3.weights", "files\yolov3.cfg")
classes = []
with open("files\coco.names", "r") as f:
    classes = [line.strip() for line in f.readlines()]
layer_names = net.getLayerNames()
output_layers = [layer_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]
colors = np.random.uniform(0, 255, size=(len(classes), 3))
```

Figure 61- Yolo Model

- Then make a function that takes the players position and ball if the ball is detected in the image (can be NULL) and graphs a circle in the position in the image in a colored circle that shows the position of the players.

```
def plane(players,ball):
    coptemp=ground.copy()
    matrix=np.array([[ 2.56945407e-01,  5.90910632e-01,  1.94094537e+02],
                     [-1.33508274e-02,  1.37658562e+00, -8.34967286e+01],
                     [-3.41878940e-05,  1.31509536e-03,  1.00000000e+00]])

    for p in players:
        x=p[0]+int(p[2]/2)
        y=p[1]+p[3]
        pts3 = np.float32([[x,y]])
        pts3o=cv2.perspectiveTransform(pts3[None, :, :],matrix)
        x1=int(pts3o[0][0][0])
        y1=int(pts3o[0][0][1])
        pp=(x1,y1)
        if(p[4]==0):
            cv2.circle(coptemp,pp, 15, (255,0,0),-1)
        elif p[4]==1:
            cv2.circle(coptemp,pp, 15, (255,255,255),-1)
        elif p[4]==2:
            #print Referee
            #cv2.circle(coptemp,pp, 15, (0,0,255),-1)
            pass
    if len(ball) !=0:

        xb=ball[0]+int(ball[2]/2)
        yb=ball[1]+int(ball[3]/2)
        pts3ball = np.float32([[xb,yb]])
        pts3b=cv2.perspectiveTransform(pts3ball[None, :, :],matrix)
        x2=int(pts3b[0][0][0])
        y2=int(pts3b[0][0][1])
        pb=(x2,y2)
        cv2.circle(coptemp,pb, 15, (0,0,0),-1)
    return coptemp
```

Figure 62- Object detection

- Then create a function that gets the players from the image and draws inside the image rectangles for each player and referee.

```
def get_players(outs,height, width):
    class_ids = []
    confidences = []
    boxes = []
    players=[]
    for out in outs:
        for detection in out:
            scores = detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > 0.5:
                # Object detected
                center_x = int(detection[0] * width)
                center_y = int(detection[1] * height)
                w = int(detection[2] * width)
                h = int(detection[3] * height)
                # Rectangle coordinates
                x = int(center_x - w / 2)
                y = int(center_y - h / 2)

                boxes.append([x, y, w, h])
                confidences.append(float(confidence))
                class_ids.append(class_id)

    indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
    for i in range(len(boxes)):
        if i in indexes:
            x, y, w, h = boxes[i]
            label = str(classes[class_ids[i]])
            if label=='person':
                players.append(boxes[i])

    return players
```

Figure 63-Draw

- Then we implement the created functions by looping in the frames of the video then changing its color to grey to make the detection on it and extract from it the players and the ball and frame them with different colors which we specify to differentiate between them.

```
opr=0
while(cap.isOpened()):
    ret, frame = cap.read()

    players=[]
    ball=[]
    if opr<310:
        opr=opr+1
        continue

    if ret == True :

        copy=frame.copy()
        gray= cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        height, width, channels = frame.shape

        blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416),
                                     (0, 0, 0), True, crop=False)

        net.setInput(blob)
        outs = net.forward(output_layers)
        outs=get_players(outs, height, width)
        for i in range(len(outs)):
            x, y, w, h = outs[i]
            roi = frame[y:y+h,x:x+w]

            #some frames are bad so resize function throw an error
            try:
                roi=cv2.resize(roi, (96,96))
            except:
                continue
            ym=model.predict(np.reshape(roi,(1,96,96,3)))
            ym=argmax(ym)

            players.append([x,y,w,h,ym])
```

Figure 64-Implementation

```
            if ym==0:
                cv2.rectangle(copy, (x, y), (x + w, y + h), (0,0,255), 2)
            elif ym==1:
                cv2.rectangle(copy, (x, y), (x + w, y + h), (0,255,0), 2)
            elif ym==2:
                cv2.rectangle(copy, (x, y), (x + w, y + h), (255,0,0), 2)


        res = cv2.matchTemplate(gray,temp,cv2.TM_SQDIFF_NORMED)
        min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(res)
        if min_val < 0.05:
            top_left = min_loc
            bottom_right = (top_left[0] + wt, top_left[1] + ht)
            ball.append(top_left[0])
            ball.append(top_left[1])
            ball.append(wt)
            ball.append(ht)
            cv2.rectangle(copy,top_left, bottom_right, (0,255,100), 2)

        p=plane(players, ball)

        out.write(copy)
        out2.write(p)

        cv2.imshow('img',copy)
        cv2.imshow('plane',p)


    if cv2.waitKey(1)==27:

        break
```

Figure 65- Implementation - 2

TOOLS AND TECHNOLOGIES

Programming Languages:

We used the **Python** programming language to prepare our data and build our machine learning models because it is flexible and practical in dealing with Data.

To build our app we used **Flutter** open source UI software because it is so flexible to develop cross platform applications.

Online Tools

Google Colaboratory, Collaborator, or “Colab” For short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python Code through the browser, and is especially well suited to machine Learning, data analysis and education.

More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

Features:

Based on Jupyter Notebook

Enable you to run Linux commands

Access the local file system

Write and execute code in Python

Free Cloud service with free GPU

Create/Upload/Share notebooks

Import/Save notebooks from/to Google Drive (Mount drive)

Import/Publish notebooks from GitHub

Import external datasets e.g. from Kaggle

Integrate PyTorch, Tensor Flow, Keras, OpenCV

Libraries

Data Analysis

Data Visualization

Data Splitting

Deep Learning Framework

Arrays and matrices calculation

Database

Firebase as our server to upload data from application and fetch data.

Models

We used Machine Learning Technology to build and train our rating, prediction and image processing models. For the rating model we used multiple linear regression algorithms and for the prediction model we used Support vector regression algorithm and for image processing we are going to use **OpenCV**, **numpy** and **tensor flow** Libraries.

Data API

Rapid API: is the **API** used to get the historical data for players we used to train our rating and prediction models.

CONCLUSION, RESULTS AND FUTURE WORK

Conclusion

In this project, fundamental cases of sports analytics were Studied (player performance Prediction) on a different leagues players from different countries after perform the data cleaning and optimization on the data got from a well-known API and along +5 past seasons and by applying a different models such as rating model which calculate the rate of each player depend on his game position (we divided them into 5 main positions) which related to some features we analyzed and choose after study a lot of previous research papers and related work and applying some methods of feature optimization and also the prediction models which depend on the rats of the player we got from the rating model over past seasons to predict the next seasons performance.

Also we started in the video analysis models and we achieved the stage of players and football detection and differentiate between the players of the two teams in addition to the referral and the ball, which will help us in the next phase to fetch the required features for the rating and prediction models instead of depend on the API data.

And finally we implemented a mobile application to act as the system interface to and integrate it with the previous mentioned models.

Results

We were able to rate the players of premier league based on our own model and to predict the result of the following season depend on the rate we calculated on the previous 4 seasons and with very high accuracy

```
Result = pd.DataFrame({'Actual Rate':y_test ,'Predicted_Rate':y_pred})
Result
```

	Actual Rate	Predicted_Rate
475	7.050000	6.970055
1762	6.650000	6.879814
1472	7.075000	7.014342
1066	6.500000	6.761418
1563	6.900000	6.934651
...	...	...
884	6.533333	6.911777
405	7.600000	7.072073
722	7.800000	6.967387
406	7.600000	7.011086
1798	7.100000	7.216424

Figure 66- Actual vs predicted

Comparing to the actual rate of the players in the year 2020.

Future Work

The results we got from the system have shown that it is possible to make long–term Predictions about player's performance.

Data from cameras and wearables would be an invaluable asset to any football analytics research.

Another idea would be the evaluation of newly promoted teams' ability and the study of their performance to comprehend what are the factors that lead them to be successful or not.

Problem with models' bias in favor of bigger clubs and difficulties in predicting draws were not fully resolved.

Fundamental differences between leagues should be specified, otherwise models could only be
Applicable on individual leagues and not become universal.

Additionally, it would be very useful if future to take into consideration some aspects that were not examined in this project such as:

- Player fatigue or starting lineup rotation due to consecutive matches

- Important player long–term injuries are factors that can affect Players or teams, but at the same time can make a model very complex. However, if the complexity is confronted, those data could be great assets for the project

- Improving the accuracy of the rating model by training it with more data with respect to avoiding overfitting.

- Improving the accuracy of the prediction model by training it with more data with respect to avoiding overfitting.

- Improving the accuracy of players and ball detection.

- Calculate Statistics from the video like pass, shoot, goals….etc.

- Improve the accuracy of the real time detection in video by training the detection model with more samples and cover many scenarios that the player may be in, for example when he is lying on the ground or injured or jumping...etc.

- Use more cameras to track the ball more often to get accurate statistics.

REFERENCES

- Sports Analytics for Football League Table and Player Performance Prediction (Christos Tjortjis, Victor Chazan-Pantzalis) Oct 2020.

- Evaluation of the Technical Performance of Football Players in the UEFA Champions League (Jan 2020).

- Assessment and Evaluation of Football Performance (Toni Graf-Baumann) Feb 2000.

- Performance indicators as a resource for the selection of talented football players https://www.researchgate.net/publication/321184385_Performance_indicators_as_a_resource_for_the_selection_of_talented_football_players

- Deep Learning for Time Series Forecasting : Predict the Future with MLPs, CNNs and LSTMs in Python by Jason Brownlee

- Data Analysis Nano Degree: https://egfwd.com/data/.

- Data Science and Machine Learning Developer Certification https://www.udemy.com/course/data-science-and-machine-learning-developer-certification/

- Tensor Flow Developer Certificate in 2021: Zero to Mastery

  https://www.udemy.com/course/tensorflow-developer-certificate-machine-learning-zero-to-mastery/

- Computer Vision Bootcamp™ with Python (OpenCV) - YOLO, SSD https://www.udemy.com/course/computer-vision-bootcamptm-python-and-opencv/

- Udemy - Flutter & Dart - The Complete Guide [2020 Edition] 2020-9 https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/

- Udemy - The Complete 2020 Flutter Development Bootcamp with Dart
  https://www.udemy.com/course/flutter-bootcamp-with-dart/

- Deep Learning Course with Flutter & Python

  https://www.udemy.com/course/flutter-deeplearning-course/.

- Rapid API-Football
  https://rapidapi.com/api-sports/api/api-football

- Machine Learning A-Z™: Hands-On Python & R In Data Science

  https://www.udemy.com/course/machinelearning/