Package 'indonesiaFootballScoutR'

February 6, 2026

Type Package

Title Tools for Football Player Scouting in Indonesia

Version 0.1.3

Description Provides tools to scrape, clean, and analyze football player data from Indonesian leagues and perform similarity-based scouting analysis using standardized numeric features. The similarity approach follows common vector-space methods as described in Manning et al. (2008, ISBN:9780521865715) and Salton et al. (1975, <doi:10.1145/361219.361220>).

License MIT + file LICENSE

Encoding UTF-8

Imports dplyr, rvest, purrr, tibble, stringr, readr, proxy

RoxygenNote 7.3.3

URL https://github.com/tioanta/indonesiaFootballScoutR

BugReports https://github.com/tioanta/indonesiaFootballScoutR/issues

NeedsCompilation no

Author Tio Anta Wibawa [aut, cre]

Maintainer Tio Anta Wibawa <tio158@gmail.com>

Depends R (>= 4.1.0)

Repository CRAN

Date/Publication 2026-02-06 20:00:13 UTC

Contents

`clean_player_db` *Clean and standardize football player data*

Description

This function converts character-based numeric fields into numeric values and prepares player data for further analysis.

Usage

```
clean_player_db(df)
```

Arguments

`df`	A data frame containing raw football player data. Must include at least columns `name`, `age`, and `market_value_est`.

Details

The function performs safe numeric conversion and does not remove rows with missing values.

Value

A data frame with cleaned and standardized player data.

Examples

```
df <- data.frame(
  name = c("Player A", "Player B"),
  age = c("21", "23"),
  market_value_est = c("€500k", "€750k"),
  club = c("Club A", "Club B"),
  league_country = c("Indonesia", "Indonesia"),
  stringsAsFactors = FALSE
)

clean_player_db(df)
```

`get_similar_players` *Retrieve similar players based on cosine similarity*

Description

Retrieve similar players based on cosine similarity

Usage

```
get_similar_players(model, player_name, top_n = 5)
```

Arguments

`model`	A trained scouting model returned by `train_scout_brain()`.
`player_name`	Character string specifying the reference player.
`top_n`	Integer indicating the number of similar players to return.

Details

Similarity is computed using cosine similarity on standardized numeric features. The reference player is excluded from the results.

Value

A data frame with similarity scores for the most similar players.

Examples

```
df <- data.frame(
  name = c("Player A", "Player B", "Player C"),
  age = c(21, 23, 22),
  market_value_est = c(500, 750, 600),
  club = c("Club A", "Club B", "Club C"),
  league_country = c("Indonesia", "Indonesia", "Indonesia"),
  stringsAsFactors = FALSE
)

model <- train_scout_brain(df)
get_similar_players(model, "Player A", top_n = 2)
```

init_real_scout *Initialize scouting workflow*

Description

This function initializes an in-memory scouting workflow. It does not create any directories or write files.

Usage

```
init_real_scout()
```

Details

This function is retained for API compatibility but performs no file system operations in order to comply with CRAN policies.

Value

NULL. Called for side effects only.

Examples

```
init_real_scout()
```

save_raw_data *Save raw scouting data*

Description

Save raw scouting data

Usage

```
save_raw_data(df, file = NULL)
```

Arguments

df	A data frame containing scouting data.
file	Optional file path. If NULL, no file is written.

Value

If `file` is provided, the file path. Otherwise, NULL.

Examples

```
df <- data.frame(
  name = "Player A",
  age = 21,
  market_value_est = 500,
  club = "Club A",
  league_country = "Indonesia"
)

tmp <- tempfile(fileext = ".csv")
save_raw_data(df, file = tmp)
```

---

scrape_club *Scrape players from a club page*

Description

Scrape players from a club page

Usage

```
scrape_club(club_url, league_country)
```

Arguments

club_url	Character string specifying the club URL.
league_country	Character string indicating league or country.

Value

A tibble containing player data for the club.

---

scrape_league *Scrape football player data from a league*

Description

Scrape football player data from a league

Usage

```
scrape_league(league_url, league_country = "Unknown League")
```

Arguments

`league_url`	Character string specifying the league URL.
`league_country`	Character string indicating league or country.

Details

This function performs web scraping and returns the data in memory. No files are written to disk.

Value

A tibble containing raw player data.

`scrape_player` *Scrape a single player row*

Description

Scrape a single player row

Usage

```
scrape_player(node)
```

Arguments

`node`	HTML node corresponding to a player row.

Value

A tibble with player information.

`train_scout_brain` *Train a similarity-based scouting model*

Description

This function prepares numeric player features for similarity-based scouting analysis.

Usage

```
train_scout_brain(df)
```

Arguments

`df`	A cleaned data frame containing player information.

Details

The returned object is intended to be used as input for `get_similar_players()`.

Value

A list containing:

data A numeric matrix of standardized player features.

players Character vector of player names.

Examples

```
df <- data.frame(
  name = c("Player A", "Player B"),
  age = c(21, 23),
  market_value_est = c(500, 750),
  club = c("Club A", "Club B"),
  league_country = c("Indonesia", "Indonesia"),
  stringsAsFactors = FALSE
)

model <- train_scout_brain(df)
```

Index