Kloppy: A Python Package for Standardizing Soccer Tracking and Event Data

Koen Vossen Joris Bekkers Pieter Robberechts
Jan van Haaren

2025-06-11

Summary

Soccer (football) data analysis has become increasingly important in modern sports, with teams, analysts, and researchers relying on detailed tracking and event data to gain insights into player performance, team tactics, and match dynamics. However, each data vendor provides information in proprietary formats with different coordinate systems, event definitions, and data structures, creating significant barriers for analysts who want to work with data from multiple sources or build vendor-agnostic analysis tools.

Kloppy is a Python package that addresses these standardization challenges by providing a unified interface for loading, processing, and transforming soccer tracking and event data from multiple vendors. The package introduces a vendor-independent data model for both event and tracking data, streamlining data preprocessing and ensuring seamless integration into data analysis and video analysis workflows. By standardizing access to soccer match data, Kloppy aims to be an essential building block for anyone working in the field soccer analytics.

Statement of Need

The soccer analytics ecosystem suffers from significant fragmentation due to the variety of proprietary data formats used by different vendors. Each vendor of soccer data uses its own unique format to describe the course of a game, meaning software written to analyze this data has to be tailored to a specific vendor and cannot be used without modifications to analyze data from other vendors. This creates several critical problems:

1. **Development inefficiency**: Analysts, researchers and data scientists must write custom parsers for each data provider, duplicating effort across the community

2. **Limited interoperability**: Models and analysis tools developed for one data format cannot easily be applied to data from other providers
3. **High barrier to entry**: New researchers and analysts face steep learning curves when trying to work with multiple data sources
4. **Reduced reproducibility**: Research findings become difficult to replicate across different datasets due to format dependencies

These challenges significantly slow down research progress and limit the practical application of soccer analytics methodologies across different data sources and organizations.

Key Features

Kloppy is implemented in Python and designed with modularity and extensibility in mind. The architecture consists of several key components:

- **Serializers/Deserializers**: Vendor-specific modules that handle the parsing and writing of different data formats
- **Domain Models**: Standardized data structures that represent soccer-specific concepts like players, events, frames, and coordinates
- **Transformers**: Tools for converting between different coordinate systems and data orientations
- **Filters**: Utilities for selecting and manipulating subsets of data based on various criteria

The package supports Python versions 3.9-3.12 and integrates well with the broader Python data science ecosystem, including Pandas (McKinney, 2010) and Polars (Vink & Polars Contributors, 2023) for data manipulation and analysis.

Data Loading and Serialization

Kloppy implements a standardized data model that can load event and tracking data from the most common data providers, supporting both public and proprietary data. The package provides out-of-the-box deserializers for all soccer data vendors shown in the table below. It handles compressed files and can load data directly from the cloud, making it flexible for various deployment scenarios.

Provider	Event Data	Tracking Data	Public Data
DataFactory	✓		
Hawkeye (2D)		✓	
Metrica	✓	✓	✓
PFF	△	✓	✓
SecondSpectrum	△	✓	

Provider	Event Data	Tracking Data	Public Data
Signality		✓	
SkillCorner		✓	✓
Sportec	✓	✓	✓
StatsBomb	✓		✓
Stats Perform	✓	✓	
Opta	✓		
Tracab		✓	
Wyscout	✓		✓

Coordinate System Transformations

Different data providers use varying coordinate systems and pitch dimensions, which can make combining datasets challenging. Kloppy flexibly transforms a dataset's pitch dimensions from one format to another (e.g., from Opta's 100x100 to Tracab centimeters to SecondSpectrum meters) and transforms the orientation of a dataset (e.g., from the home team and away team direction of play each period (`Orientation.HOME_AWAY`) to the home team playing left to right the whole game (`Orientation.STATIC_HOME_AWAY`), to every attacking playing out from left to right (`Orientation.BALL_OWNING_TEAM` and `Orientation.ACTION_EXECUTING_TEAM`)). Additionally, Kloppy supports user-defined custom coordinate systems. These transformations enable seamless data integration and consistent analysis across multiple sources.

Pattern Matching and Event Search

Kloppy provides the ability to search for complex patterns in event data, enabling users to identify specific tactical sequences or game situations efficiently. The package implements a powerful search mechanism that combines regular expressions with graph-based pattern matching using NetworkX (Hagberg et al., 2008), enabling users to find tactical moments more quickly and easily.

Standardized Data Models

The package implements comprehensive data models for both tracking and event data that abstract away vendor-specific details while preserving essential information. The core data structures include:

Event Data Model

Each Event is associated with an `EventType` which classifies the general event type that has occurred. Furthermore, each `Event` consists of general attributes (e.g. player, team, timestamp, period) as well as event type-specific attributes. Each event can also have a list of `Qualifier` entities providing additional information about the event (e.g. set piece type, card type, pass type or body part).

The table below shows and overview of all event types, and their provider specific support as provided by Kloppy.

Event Type	StatsBomb	Stats Perform	Wyscout v2	Wyscout v3	DataFactory	Sportec	M
Generic	✓	✓	✓	✓	✓	✓	✓
Pass	✓	✓	✓	✓	×	✓	✓
Shot	✓	✓	✓	✓	×	✓	✓
TakeOn	✓	✓	×	✓	×	×	✓
Carry	✓	×	×	×	×	×	✓
Clearance	✓	✓	✓	✓	×	×	×
Interception	✓	✓	✓	✓	×	×	×
Duel	✓	✓	✓	✓	×	×	×
Substitution	✓	×	×	×	×	✓	×
Card	✓	✓	✓	✓	×	✓	×
PlayerOn	✓	×	×	×	×	?	?
PlayerOff	✓	×	×	×	×	?	?
Recovery	✓	✓	✓	×	×	?	✓
Miscontrol	✓	✓	✓	×	×	?	×
BallOut	✓	✓	✓	×	×	?	○
FoulCommitted	✓	✓	✓	✓	×	✓	✓
Goalkeeper	✓	✓	✓	✓	×	?	×
Pressure	✓	×	×	×	×	×	×
FormationChange	✓	✓	×	✓	×	×	×

Legend: ✓ = Full support, ○ = Partial support, × = Not supported, ? = Unknown

The table below shows and overview of all possible qualifiers.

Qualifier Type	Values
PassQualifier	CROSS, HAND_PASS, HEAD_PASS, HIGH_PASS, LAUNCH, SIMPLE_PAS
BodyPartQualifier	RIGHT_FOOT, LEFT_FOOT, HEAD, OTHER, HEAD_OTHER, BOTH_HA
CardQualifier	FIRST_YELLOW, SECOND_YELLOW, RED
CounterAttackQualifier	-
DuelQualifier	AERIAL, GROUND, LOOSE_BALL, SLIDING_TACKLE
GoalkeeperQualifier	SAVE, CLAIM, PUNCH, PICK_UP, SMOTHER, REFLEX, SAVE_ATTEMP
SetPieceQualifier	GOAL_KICK, FREE_KICK, THROW_IN, CORNER_KICK, PENALTY, KI

Tracking Data Model

The tracking data architecture centers around the `Frame` class, which serves as a temporal snapshot containing all positional information for a single moment in time. Each frame maintains a `frame_id` for unique identification, `ball_coordinates` (represented as 3D points), and a `players_data` dictionary that maps `Player` objects to `PlayerData` instances. The `PlayerData` class encapsulates not only coordinate positions but also derived metrics like distance traveled, instantaneous speed, and an `other_data` dictionary. The `TrackingDataset` class aggregates these frames along with essential metadata including frame rate, coordinate system information etc. This hierarchical structure maintains player identity consistency across frames while supporting efficient queries like `players_coordinates` that provide quick access to positional data.

Metadata

The `Metadata` class serves as the central repository for all contextual information associated with a dataset, cleanly separating structural metadata from the actual data records. This class encapsulates essential game-level information including `home_team` and `away_team` references, temporal structure through `periods` (containing start/end timestamps and period identifiers), and critically, the `coordinate_system` that defines how spatial data is represented. The coordinate system component tracks pitch dimensions, orientation (e.g., fixed vs. team-relative), and origin points, enabling seamless transformation between different vendor coordinate spaces. Additionally, the metadata maintains frame rate information for tracking data, data provider information, etc.

Player and Team

Kloppy implements comprehensive identity management through standardized `Player` and `Team` classes that maintain consistency across different data providers. The `Player` class encapsulates essential attributes including the provided `player_id`, `jersey_no`, player names (`first_name`, `last_name`), and crucially, a `starting_position` field that uses Kloppy's standardized `PositionType` enumeration. This position system provides a hierarchical classification (e.g., `LeftCenterBack` is a subtype of `Defender`) that abstracts away provider-specific position nomenclature. The `Team` class maintains team identity through `team_id`, official `name`, `ground` designation (home/away), and a `players` list containing all team members. Teams also include formation information through `starting_formation` and a dynamic `formations` attribute that tracks tactical changes throughout the match using a `TimeContainer` structure.

Impact and Applications

Since its initial release in 2020, Kloppy has been adopted by researchers, analysts, and organizations working with soccer data. The package has enabled:

- **Research reproducibility**: Studies can now be more easily replicated across different data sources

- **Rapid prototyping**: Analysts can quickly test ideas across multiple datasets without rewriting data loading code
- **Educational accessibility**: Students and newcomers to soccer analytics can focus on learning analytical techniques rather than data parsing
- **Industry adoption**: Organizations can build more flexible analysis pipelines that aren't locked to specific data vendors

The standardization provided by Kloppy has particular value for the growing field of soccer analytics research, where the ability to validate findings across multiple data sources is crucial for scientific rigor.

Acknowledgments

Kloppy is supported by PySport, a non-profit organization (RSIN: 866294211) dedicated to advancing sports analytics through open-source software. The project welcomes contributions from the community, including bug reports, feature requests, and code contributions.

The authors thank the PySport community and all contributors who have helped develop and improve Kloppy. We also acknowledge the various data providers who have made public datasets available, enabling the development and testing of standardization tools like Kloppy.