CONFIDENTIAL

driblab

FOOTBALL POWERED BY DATA

Driblab - Capstone Task
xG Model with Tracking Data

May 2025

PRESTIGE. ACCURACY. RELIABILITY

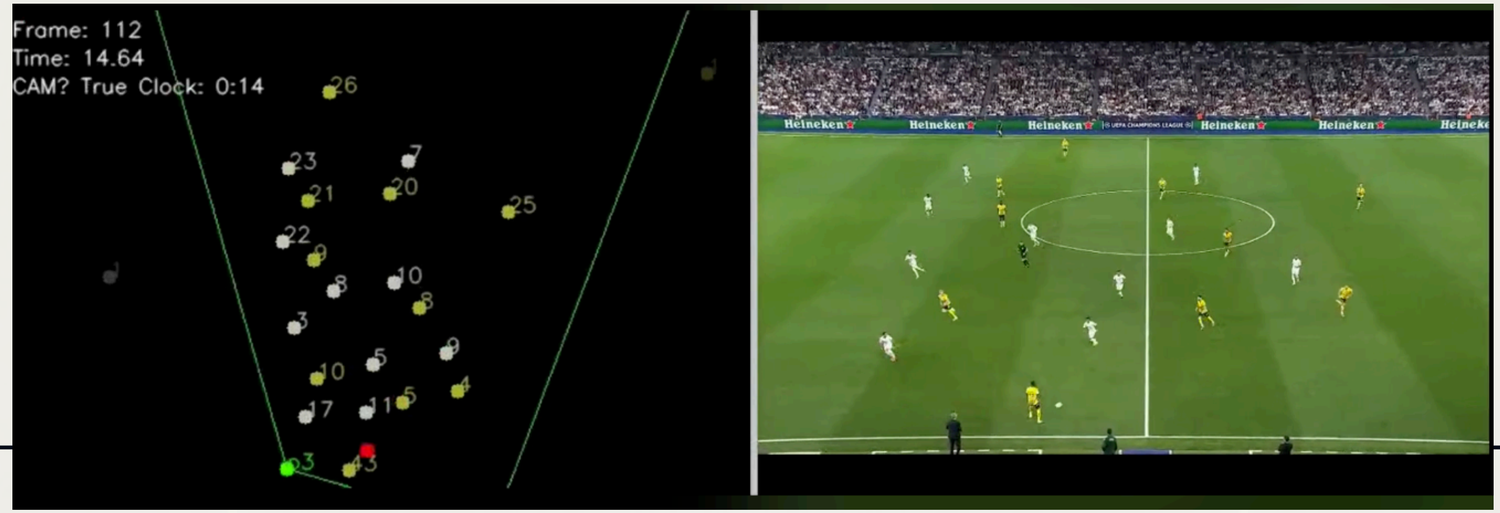
BACKGROUND INFORMATION

For this Project, you will be tasked with developing an Expected Goals (xG) model that can utilise tracking data to improve the accuracy of its predictions.

Traditional xG models use event data, formatted in a tabular manner to predict if a goal is scored or not. An example of this can be seen below:

X & Y coords.	Angle from Goal	Distance from Goal	Body Part Used		Goal Likelihood
(20, 40)	84	10	Foot	➡	0,46

Tracking Data on the other hand contains the coordinates of all players on the pitch, collected with some regular frequency. An example of the tracking data is visualised below, alongside the actual frame from the broadcast footage:



DIRECTORY STRUCTURE

The data will be provided to you as a zip file called shot_pack.zip and can be accessed from this link. Once unzipped the directory will have the following structure

```
/
|__shots/
    |_____661620.json
    |_____663910.json
      …
|__jsonls/
    |_____661620_tracking_data.jsonl
    |_____663910_tracking_data.jsonl
    …
```

The information about each shot can be found in the shots directory, where the file's name indicates the match_id of the game. Within the jsonls directory you can find the tracking data for the same game.

driblab

DATA FORMAT - SHOTS

For each match_id the file will contain an array of JSON Objects that describe the details of a shot. This is also called the event data of the shot.

The matchId within each shot event data is not the same as the match_id used in the naming of the file and the corresponding tracking data file.

An example of some of the fields present in such a shot can be seen here:


```
{
    "id": 2306655127,
    "matchId": 5588915,
    "matchTimestamp": "00:11:43.702",
    "videoTimestamp": "704.702784",
    "relatedEventId": null,
    "type": {
        "primary": "shot",
            "secondary": [
            "opportunity",
            "Touch_in_box"
            ]
    },
    "location": {
        "x": 89,
            "y": 31
    },
    "player": {
        "id": 559712,
            "name": "E. Colley",
            "position": "LAMF"
    },
    "pass": null,
    "shot": {
        "bodyPart": "right_foot",
            "isGoal": false,
        "onTarget": true,
            "goalZone": "gb",
        "xg": 0.08024,
        “xg2”: 0.0722,
            "postShotXg": 0.07932,
        "goalkeeperActionId": 2306655350,
            "goalkeeper": {
            "id": 7848,
            "name": "E. Martínez"
            }
    },
      ...
```


driblab

DATA FORMAT - TRACKING DATA

For each tracking data file, the first line contains information about the game and players. It serves as a look-up dictionary to map player IDs to player information.

This approach avoids unnecessary repetition and reduces file size. An example is shown here:


```
{
 "match_data": {
  "date": "2024-03-01",
  "match_id": 553262,
  "result": {
   "home": 0,
   "away": 1
  },
  "season_data": {
   "id": 816545,
   "name": "POL I 2023"
  }
 },
 "players_data": {
  "team0_id": {
   "player0_id": {
    "name": "Name Surname",
    "number": 10,
    "position": "DC"
   },
   {...}
  },
  "team1_id": {
   "player0_id": {
    "name": "Name Surname",
    "number": 10,
    "position": "DC"
   },
   {...}
  }
 },
 "fps":10
}
```


driblab

DATA FORMAT - TRACKING DATA

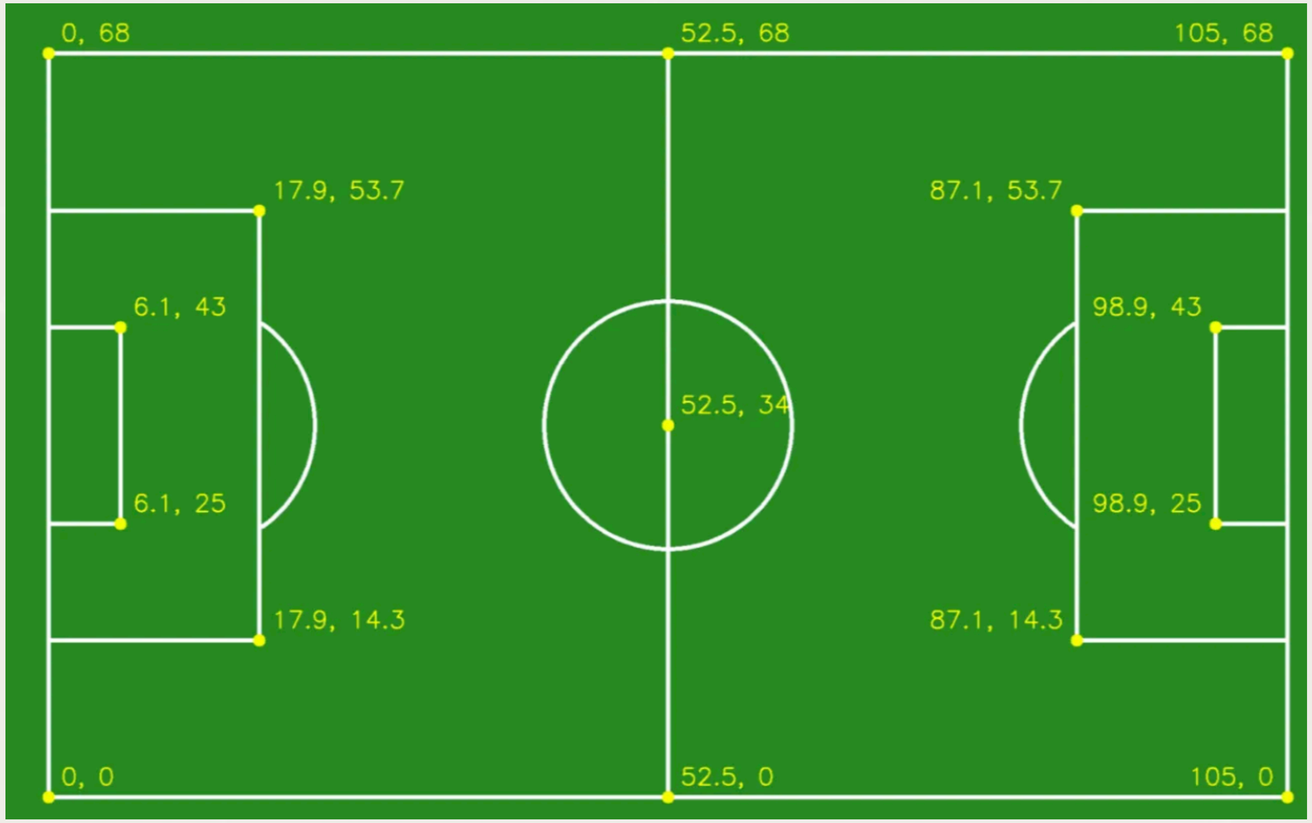
After the first line, all subsequent lines follow a common structure. This structure is straightforward, and an example is provided here:

```
{
  "frame": number,
  "vid_timestamp": number, // Seconds elapsed in original video
  "period": number,
  "ball": [x, y] // The x and y coordinates of the ball
  "data": {
    _id: [ // Array of players that play for team0_id
      {
        "id": number // Corresponds with look-up dict
        "x": number // The x coordinate of the player
        "y": number // The y coordinate of the player
        "vis": boolean
      },
      { ... } // Remaining players for team0_id
    ],
    team1_id: [
      ... // The same structure for the players of team1_id
    ]
  },
  "cam": [ // Polygon describing broadcast camera view
    [x0,y0], // The first coordinate
    [x1,y1], // The second coordinate
    [x2, y2],// The third coordinate
    [x3, y3] // The fourth coordinate
  ]
}
```

driblab

DATA FORMAT - TRACKING DATA

The coordinate system used can be visualised as follows:



driblab

ALIGNING THE DATASETS

In order to include tracking data when trying to calculate the xG of a shot, an alignment step must be performed for each shot to find the corresponding tracking data frame. This can be done by utilising the videoTimestamp field in both datasets as they are aligned.

Thus, to find the corresponding tracking data frame for the shot provided in the previous example, you would need to find the value of the videoTimestamp field. In this case it would be this object within the event data:
… "videoTimestamp": "704.702784" …

You would then need to find the corresponding <match_id>_tracking_data.jsonl file, and find the frame that has the closest videoTimestamp value. Thus you would have created an event data & tracking data pair.

THE TASK

For each shot, you will be provided with two xG values, namely xG, and xG2. These can be found within each shot's event data and the outputs represent two different xG models that were trained using only the event data, i.e without the context of the surrounding players. Your job is to use these two models as a baseline and attempt to use the information provided with the tracking data frame to obtain a more accurate xG model. Do not use these xG values as features for your xG model.

You will do this by performing feature engineering on the event data to create features that you think are relevant to explaining the likelihood of a goal being scored, such as the angle & distance from goal, body part used, etc.

You will then carry out the same process with the tracking data, by crafting features based on the coordinates of the players, such as the position of the goalkeeper, if there were any players between the shooter and the goal, how close the nearest opposition players were, etc.

THE TASK

Once you perform the feature engineering on both types of data and you collect them into one feature vector for each shot, you will then try to use this information to predict whether or not a goal was scored. Thus the dataset should be structured that a goal is represented by a value of 1 and any other outcome for a shot is represented by a 0. The outcome of each shot can be obtained from the event data.

It is important that you carry out all the steps expected in a machine learning project and utilise a training and test set as well as identifying the right metrics to use, and comparing your new model's findings with the baseline models. Ensure that there is no data leakage between train and test sets.

DELIVERABLES

You should provide clearly structured and commented code detailing your experiments, as well as a report that explains your findings. The report should not be a documentation of the code.

Please ensure that you outline the key technical decisions that were taken, such as the type of model used and the feature engineering steps. Also include the comparisons between the baseline models and your final model.

It would also be interesting if you could identify which event data features are most informative? Which tracking features add the most value over event-only xG?

driblab

CONTACT

driblab

DRIBLAB WAS FOUNDED AS A BOUTIQUE CONSULTANCY FOR FOOTBALL EXECUTIVES IN 2017. TODAY IT IS THE ONE-STOP-SHOP FOR THE ENTIRE DATA AND FOOTBALL UNIVERSE, HAVING DEVELOPED AD-HOC PROJECTS FOR MORE THAN 100 CLIENTS WORLDWIDE.

WE ARE HEADQUARTERED IN MADRID, SPAIN AND WE ARE PARTNERS OF THE WORLD FOOTBALL SUMMIT AND IE BUSINESS SCHOOL.

OUR SERVICES HAVE BEEN REFLECTED IN MEDIA SUCH AS SUNDAY TIMES, BBC, FORBES, EL PAÍS, RTVE, DIARIO MARCA, LA VANGUARDIA, EL CONFIDENCIAL OR ESPN.

ADMINISTRATION

admin@driblab.com

HEAD OF TECHNOLOGY

Cristian Ramiro

ccramiro@driblab.com

ADDRESS

PASEO IMPERIAL 6. 2B
MADRID, 28005 SPAIN