ELEN0062 - Introduction to Machine Learning
Project 3 - Competition
Football pass prediction

November 20th, 2025

The aim of this project is to get some experience with designing a solution to a complex problem on real data, using the different tools seen in the course. You will be able to compare the performance of your solution with other groups on a leaderboard during this competition, although it is obviously not the sole grading factor.

The code and data can be found on the projects website: `https://iml.isach.be`. As for the first two projects, the submissions will be done on Gradescope[1]. The project must be carried out by groups of at most three students. See below for the deadlines.

Goal

This project lies within the field of sports analytics. The project objective is to predict the recipient of a pass during a football match based on information about the positions of all players on the pitch. Potential applications include for example highlighting tactical decisions made by players and enhancing broadcast experiences by helping automated camera systems anticipate ball movements.

Data

The full dataset contains 11686 samples collected during 14 different games involving a Belgian football club during the 2014/2015 football season. Each sample (corresponding to a pass) is a snapshot of the situation when the ball is passed, and contains the following features (46 in total):

- `time_start`: the time at which the pass takes place (in ms, since the start of the half)
- `sender_id`: the ID of the player (sender) with the ball. Each player is identified by an integer ranging from 1 to 22.
- `x_1,y_1,...,x_22,y_22`: the positions of the 22 players on the field (11 per team). Players 1 to 11 are the players from the home team, while players 12 to 22 are the players from the away team. The coordinates are given in cm, with the center of the pitch at the position (0,0). The pitch is 105 meters long (x-axis) and 68 meters wide (y-axis), meaning that x and y coordinates within the pitch are respectively in the intervals $[-5250, 5250]$ and $[-3400, 3400]$.

The full dataset has been randomly divided into 8686 samples to train your model and a test set of 3000 samples to assess it. The ID of the receiver, which, as for the sender, ranges from 1 to 22, serves as target and is only provided for the training set examples.

The available data is splitted into three files in the CSV format:

- `input_train_set.csv`: the input data of the training set
- `input_test_set.csv`: the input data of the test set

[1] `https://www.gradescope.com`, Entry code: 7XYGVW.

- `output_train_set.csv`: the output data of the training set

Each line of these files start with an ID of the pass (ranging from 0 to 8685 for the training set and from 0 to 2999 for the test set) that should obviously not be used as inputs for training the model.

Predictions and evaluation metrics

Your main goal is to predict the receiver of each pass in the test set, i.e. a ID from 1 to 22. The evaluation metric that will be used to evaluate the quality of this prediction is the classification accuracy, Acc_{TS}, i.e. the percentage of passes for which the prediction is correct. This will be the main evaluation metric used to rank your submissions during the challenge.

In addition to the receiver, we ask you also to provide an estimation of the probability of each player to receive the ball. Let us denote by $y_{i,j}$, a variable equal to 1 if player j is the receiver of the i-th pass in the test set, 0 otherwise and let us denote by $\hat{p}_{i,j}$ your estimation of the probability that player j is the receiver of the i-th pass. The quality of your probability predictions will be evaluated using the Brier score[2]:

$$BS_{TS} = \frac{1}{N_{TS}} \sum_{i=1}^{N_{TS}} \sum_{j=1}^{22} (y_{i,j} - \hat{p}_{i,j})^2,$$

where N_{TS} is the number of samples in the test set. This score is equivalent to the accuracy when the predicted probabilities are enforced to be either 0 or 1.

As a third task, we ask you also to provide an estimation of the accuracy of your model on the test set, denoted $\hat{Acc}_{TS}$. At the end of the competition, we will assess your model with a third evaluation metric, denoted Acc_{TS}^{corr}, and defined as:

$$Acc_{TS}^{corr} = Acc_{TS} - |\hat{Acc}_{TS} - Acc_{TS}|.$$

This metric is high when both the accuracy of your model on the test set is high and your estimation of this accuracy is close to the observed one.

Submission and competition

The goal is to get the best possible predictions on the test set. During the competition, your predictions can be submitted multiple times on Gradescope, where a leaderboard will allow you to compare yourself with the other groups. When submitting *during the competition*, the accuracy and Brier scores will be computed on 25% of the total test set (750 passes). We will not disclose which subset this is. All group predictions will be ranked in the leaderboard according to the accuracy. *Once the competition is over*, all three metrics will be computed using the other 75% of the test set. The private accuracy scores will count as the final scores for the competition, whereas the other ones will be used for the grading process.

Submission files to be submitted on Gradescope are CSV files that should contain 24 columns:

- `Id`: the identifier of the pass in the test set (an integer ranging from 0 to 2999).
- `Predicted`: the prediction of the player receiving the ball (an integer ranging from 1 to 22).
- `P_<ID>`, where `<ID>` ranges from 1 to 22: the probability that player `<ID>` receives the ball.

The file should contain a header and have the following format:

```
"Id","Predicted","P_1","P_2",...,"P_22"
"Estimated",0.1,0,0,...,0
0,2,0.01,0.6,...,0.0
....
2999,22,0.0,0.01,...,1.0
```

[2] `https://en.wikipedia.org/wiki/Brier_score`.

The first line of the header shows the column names, while the second line is used to provide your estimation of accuracy (in the example, 0.1). The next 3000 lines give your predictions for all test set passes. They do not need to be ordered according to the pass identifier but all 3000 passes should be present.

Important note: Submissions are limited to *3 per group per day.* This choice is deliberate, in order to avoid overfitting the public test set. As the *private* score is the one that counts and is computed after the end of the competition, you have no interest in overfitting the public test score. We ask you to respect this rule, even if Gradescope does not entirely enforce it. All submissions must be conscientiously done on behalf of all the members of your group. Note that you *can* submit past the limit, but these submissions will not count and show on the public leaderboard. However, they could count for the private leaderboard! This might be useful in case you run out of submissions on the last day. **The competition will end on December 14th at 23:59.**

Example script

To help you start, we provide a "naive" `toy_example.py` script that goes through the following steps:

- Loading the training set (both input and output files);
- Deriving features for each pair of (sender, potential receiver) (This step is only one way of addressing the problem. We strongly recommend to also consider other approaches than the one provided here);
- Making random/naive predictions (see below for more details);
- Creating a submission file following the guidelines provided in the previous section.

In particular, please note that:

- It creates a new sample set where each sample is a pair of players: the sender and player j with $j = 1, \ldots, 22$ (including the sender).
- It computes two features: the distance between the two players and if they belong to the same team.
- The `write_submission` function can make a submission with only the predicted player ID (predictions of size (`n_test_samples, `)) and/or predicted probabilities (probas of size `(n_test_samples,22)`). If only the predictions are provided, then the probabilities are derived (i.e., 1.0 for the predicted player in predictions and 0.0 for all others). If only the probabilities are given, then the predictions are derived (i.e., the predicted player is the one with the largest probability). Note that if several players have the same probability, the one with the smaller Id is selected.
- `predicted_score` is hard-coded. Do not forget to modify this value.

The `toy_example.py` script generates two submission files: one based on (randomly) predicted player ids, and one based on player probabilities predicted by a decision tree.

Report

By **December 19th**, you should have submitted a report along with your code on Gradescope.

Your report should describe your investigations on the data and on different approaches, along with your final approach and results. It *must* contain the following:

- A detailed description of all the approaches you have investigated.
- A detailed description of your approach to select and assess your model.
- A table summarising the results of your different approaches.
- All tables, figures and results should be analysed in-depth while avoiding unnecessary redundancies.

- Any complementary information that you want to analyse.

Please note that the methodology and the quality of your report matter a lot. Great results on the leaderboard with poor motivations, explanations, and details in the report are definitely not what you should aim for.

Rules

You have to *strictly adhere* to the following rules:

- Please recall the rule of 3 submissions per group per day. This is not easily enforced through Gradescope, but we ask you to respect this rule and submit conscientiously.
- You can use any techniques or algorithms you want.
- Solutions can be implemented in any language using any libraries. The only condition is that your solution should be strictly reproducible.
- You can not use external data, unless you get our approval first.
- If you use external code for algorithms, you have to give the references (even for tools or algorithms used or developed in previous projects such as Scikit-Learn).
- You can not use ready-made solutions for the problem, i.e. any available software specifically designed to solve the problem of football pass prediction.
- The use of AI tools (chatGPT, Copilot...) is not prohibited, although not encouraged either. You are expected to understand what you are doing and to strictly follow the ULiège AI Charter.
- Make sure you acknowledge and properly reference all the external sources you have used, such as algorithms, software, scientific papers and AI tools. Failure to do so will be considered plagiarism.

Bon travail!