Data Science Challenge

Nordeus - Job Fair 2026

About the challenge

A typical Data Science project involves multiple phases, including identifying business opportunities, formalizing the problem, gathering and processing data, developing and training the model, and integrating the solution into a production environment. For this assignment, we have defined the problem and prepared the data, and your task is to develop and train an ML model that will use the available data to solve the task and provide relevant predictions.

In this task, we focus on Associations (Clans) in Top Eleven and their weekend tournaments. Each association consists of 6 managers who compete against 6 managers from a rival association in a series of direct matches. Success depends not only on individual skill but also on the collective strength, activity, and preparation of the entire group. By analyzing historical performance and member attributes, we can predict tournament outcomes and provide actionable insights. Your goal is to predict which association will emerge as the winner and to identify key areas where a team can improve to tip the scales in their favor.

Data description

There are four datasets:

- member_stats_training.csv
- clan_matches_training.csv
- member_stats_test.csv
- clan_matches_test.csv (same as training dataset but without target variable)

You can find the data description below:

Variable	Definition
user_id	Unique manager number.
clan_id	Unique identifier for the association (clan).
registration_platform_specific	Platform through which the user had their first session (registration).
registration_country	Country of origin of the first session (registration).
cohort_day	Number of days since the user registered.
days_active_last_28_days	The number of distinct dates, out of the previous 28 days, where the user had at least one login.
days_active_last_7_days	The number of distinct dates, out of the previous 7 days, where the user had at least one login.
days_since_last_active	Days passed since the user had the last login in the game.
training_count_last_28_days	Total number of training sessions in the last 28 days.
is_payer_lifetime	Indicates if the user has ever made an in-app purchase.
dynamic_payment_segment	Segmenting users based on their purchase recency, frequency, and monetary value
clan_multiplier	Individual bonus multiplier based on League competition rank in the previous season
avg_stars_top_11_players	Average quality (stars) of the manager's best 11 players.
avg_stars_top_3_players	Average quality (stars) of the manager's best 3 players.
avg_training_bonus	Average level of team training bonuses (Attack, Defense, Possession, Condition).
clan_1_id	ID of the first association in the match.
clan_2_id	ID of the second association in the match.
clan_1_points	Total points scored by the first association.
clan_2_points	Total points scored by the second association.
clan_winner	**Target variable**: 1 if clan_1 won, 2 if clan_2 won.

Technical Note: Tournament Matches and Clan Multiplier Logic

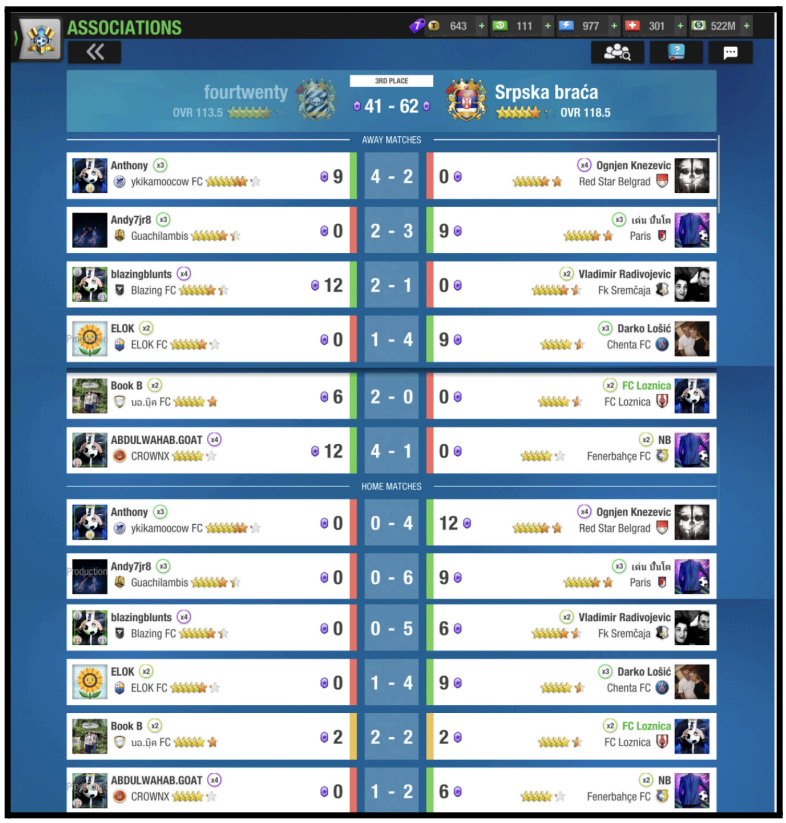
The **clan_multiplier** is a core mechanic of Association tournaments that determines the weight of each manager's contribution to the team's total score.

- *Tournament Matches*: There are a total of 12 matches (each manager plays twice: one home and one away). By default, managers from the two associations play against each other based on their team quality (the best manager from one association plays against the best manager from the other, the worst against the worst, and so on). However, managers of the association, when they play Home matches, have the option to change this default match schedule.

- *How Clan Multiplier is obtained*: A manager's League performance from the previous season determines their individual multiplier for the current season. Higher-ranked in League results in a higher multiplier.

- *How Clan Multiplier affects the game*: During an Association tournament, the base points a manager earns for their individual match (3 points for a win, 1 point for a draw, and 0 points for a defeat) are multiplied by their individual clan multiplier. These multiplied points from all 6 members are summed up to determine the final score of the Association in the match against the rival clan (On the image below you can see an example)

For example, in the first match manager Anthony won the match against manager Ognjen. The result was 4:2. Since Antony has x3 multiplier, he got 9 points in total.



Individual clan multiplier: x4 x3

Association points:

Manager's team quality:

EVALUATION:

Target variable

The target variable is **clan_winner**. It is an integer (1 or 2) describing which association won the tournament match.

What is your goal? 🎯

Goal: Predict the winning association (clan) in the test dataset

Evaluation Metric 📉

Submissions are evaluated based on the **Accuracy** between the predicted winner and the actual winner of the matches.

Bonus task 🤖

Once you've built a robust model for predicting clan tournament match outcomes, take the next step: design a "smart" assistant! This assistant will provide actionable advice to the Association on how to boost their chances of victory. For example, should managers focus on increasing training bonuses or investing in overall team quality? And is this improvement for all managers in association, or just a few managers? You have complete creative freedom to design an advisory system! On a few examples, you can show how it works and how it can be applied.

Submission format 📃

- You should save your results in the “clan_winner_predictions.csv” file.
- The submission file should contain a row for each match in the test dataset and 3 columns: clan_1_id, clan_2_id and predicted_clan_winner.

```
clan_1_id, clan_2_id, predicted_clan_winner
clan_469, clan_470, 1
clan_471, clan_473, 2
etc.
```

- The submission should be sent via email to jobfair@nordeus.com with a link to your GitHub repository (email subject: **Data Science Challenge**). Please add your **full name** to the email, as well as the link to your **LinkedIn profile**, if you have it.
- Besides the file with predictions, the repository should contain scripts/notebooks/analysis with code that shows how predictions were made. For this challenge, you can use the language of your choice, preferably Python or R.

The challenge is open until May 3rd, 2026, at the end of the day. Good luck!