Challenge data QRT: Football, who will wins ?
MVA 2023/2024

Akedjou Achraff Adjileye akedjou-achraff-brad.adjileye@universite-paris-saclay.fr
Jordan Momo Jupou jordan.momo-jupou@student-cs.fr

March 21, 2024

1 Introduction

In this challenge, our aim was to predict the outcome of a league football match. To accomplish this, we utilized four datasets grouped in two types containing various statistics. The first type of datasets refers to team statistics, while the second focuses on player statistics. These datasets quantify player and team performance in different ways: offensive features such as the number of goals, assists, shots, etc., defensive features including the number of blocks, interceptions, yellow cards, etc., as well as open play data such as passes, dribbles, etc. All values were transformed for every feature, and match identification features like the league and the team names of the players were deleted from the final evaluation data to prevent cheating.

Each match sample consists of two vectors of statistics, one for the home team and one for the away team. The players dataset contains information about all the players in the squad of each team for the match

2 Description and important remarks on the dataset

As underlined in the previous section, datasets for this challenge were meticulously transformed, rendering any attempts to "guess" the corresponding football match impossible. However, there is no information provided regarding the date of the match or the corresponding season.

It's important to note that two identical teams (by name) in the training dataset should not be perceived as identical samples for prediction. This is because the two corresponding matches could occur in different seasons or at different times within an unknown season. For instance, there are matches involving the Toulouse Football Club in both the French Ligue 1 and the French Ligue 2. Similar variations can be observed for players; for example, a squad for a Barcelona game may include the outstanding Luis Suarez of the 2015-2016 season, as well as the underperforming Luis Suarez of the 2019-2020 season.

All raw count features are presented in six formats: the sum, average, and standard deviation for all matches of the fixture teams since the start of the season, and the sum, average, and standard deviation for the last five matches of the fixture teams. However, for certain features such as the number of wins/draws/losses (WDL), only the sum is provided.

The target feature consists of three categories: win, loss, and draw, making the challenge a 3-class classification problem. The performance metric utilized is accuracy.

2.1 Missing Data

The datasets provided for the challenge contain missing data for both the training and testing sets, whether for team or player data. The figure 1 illustrates the histogram of the proportions of missing values for the team dataframes.

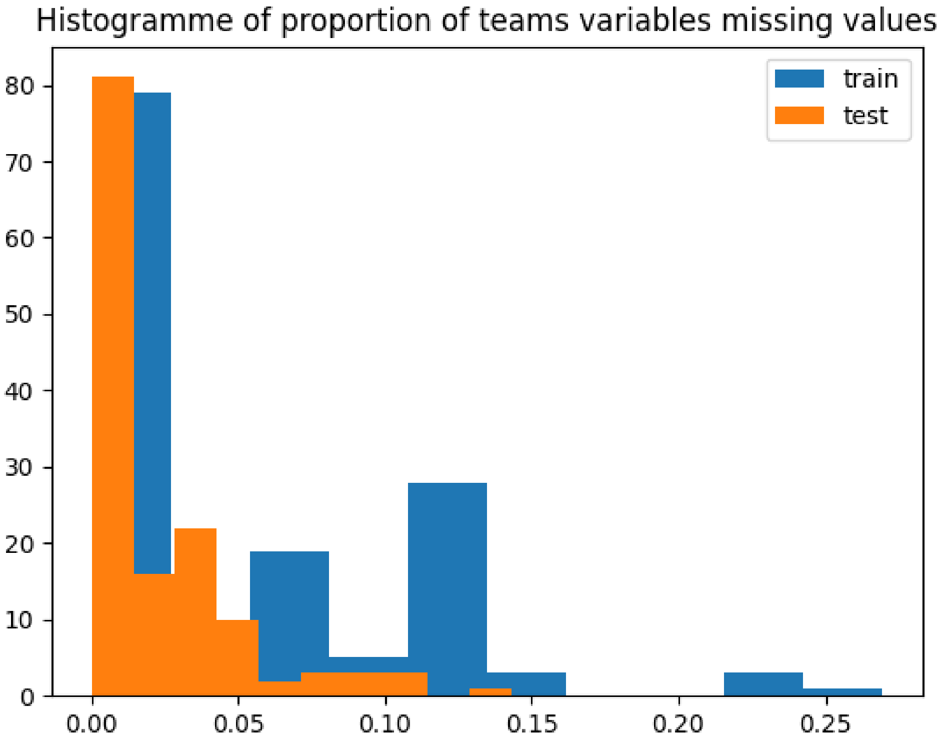

Figure 1: Similar Distribution of NaN Data Proportions in the Training and Test Datasets

Most features in the test data contain less than 5% missing values. In the training data, we observe a second peak at $11 - 12\%$, representing around thirty features.

In general, predicting the outcome of a football match is very challenging. Utilizing data provides a new approach to do so, and having missing data, especially for features that can be crucial such as WDL (wins/draws/losses) in the teams recent matches or statistics like teams goals, teams shots, teams saves (for the goalkeeper), etc., is undesirable. Therefore, we have taken a closer look at the training features that contain more than **10%** of missing values (table 1).

Upon observing the features involved and relying on our prior knowledge of football, we think that the only features that could significantly impact predictions are **Shots Inside Box** (which often leads to goals) and **Injuries** (which can significantly affect a team). It's worth noting that we have the feature **Shots Total**, so there's no need to include **Shots Outside Box**, as it's the difference between his two pairs.

We decided to fill in missing team data for all features related to the average values of all teams worldwide. The assumption is that overall, there are matches in a season where a team performs

Table 1: Training Missing Data: features with Over 10% Missing Values in 'Sum' Format, Consistent Proportions for Average and Standard Deviation

Stats	proportion
Shots Inside Box (season)	0.1126
Shots Outside Box (season)	0.1127
Passes (season)	0.1149
Successful Passes (season)	0.1125
Injuries (season)	0.1593
Shots Inside Box (5 last matchs)	0.1128
Shots Outside Box (5 last matchs)	0.113
Passes (5 last matchs)	0.1239
Successful Passes (5 last matchs)	0.1127
Injuries (5 last matchs)	0.2383

similarly to any average team, regardless of its level, form, or opponent. We consider matches with missing data as such matches. To estimate the averages of all teams worldwide, we simply calculate the means of the features across all our datasets, which are assumed to be sufficiently representative of the match distributions.

The same idea was applied to fill in missing data in player datasets. However, since the rows in this dataset represent players, the averages are calculated per team. For features specific to goalkeepers, the averages are calculated only for goalkeepers of the same team. However, there may still be missing values in cases where we have no data for any player of a team for a certain feature. In this case, we consider that this match can be seen as a match where the team in question has an average performance like any other team worldwide, and we replace it with the mean values of all teams worldwide, estimated by a mean accross all our datasets that we consider sufficiently representative.

3 Baseline

A football match is one of the most complex events to predict the outcomes of. There's a deterministic component, but also a non-negligible random aspect to what can happen during the game.

Based on our domain knowledge, we argue that in most cases, the recent performance of a team provides valuable information about the potential outcome of their next match. This performance is simply calculated by considering the number of wins/draws/losses (WDL) of the team in their last N matches (typically five). A team with a strong winning streak is considered to be in good form and has a higher chance of winning their next match compared to a team on a poor streak. However, it's important to note that this is a very simplistic hypothesis and should not be taken as a certainty because, as mentioned earlier, there is a significant random element in football games. Every scenario can suddenly unfold, adding to the charm of the sport. Nonetheless, this hypothesis allows us to establish a reasonable baseline.

We utilized the data on the number of WDL in the last five matches for each team involved in a match to compute a performance indicator called the **number of points** (N_p). This indicator is calculated using the following simple formula, inspired by professional football leagues:

$$N_p = 3 \times N_w + 1 \times N_d + 0 \times N_l$$

where N_w, N_d, and N_l represent the numbers of wins, draws, and losses respectively. We then compare the teams' number of points based only on their last five matches to predict the outcome: the team with the most points is predicted to win, and a draw is predicted if the two teams have the same number of points. This straightforward and explainable heuristic achieved an accuracy of approximately **43%** on all the training data. This performance is relatively promising for a baseline, considering that it is approximately **10% better** than chance (which would be 33% accuracy). We'll later see that we are unlikely to surpass this baseline, just as it outperforms chance. Figure 2 illustrates the results of the baseline model.

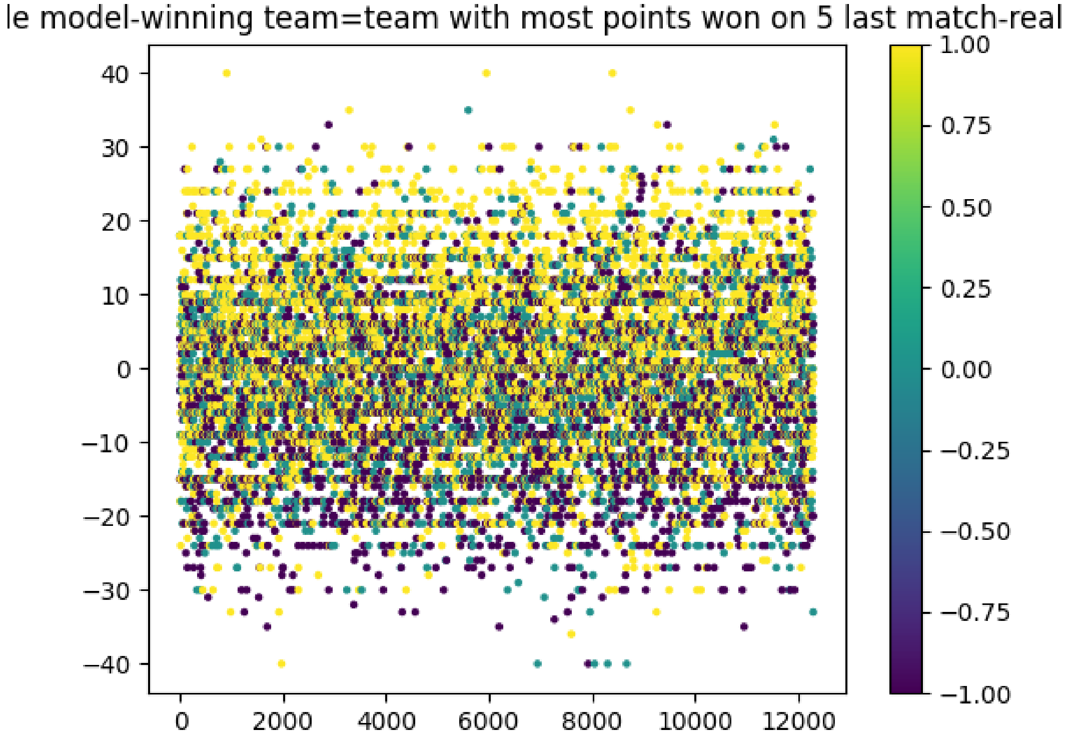

Figure 2: Baseline model predictions: yellow points indicate a home team win, dark purple points indicate an away team win, and green points indicate a draw. On the y-axis, we have the difference in the number of points: $N_p(\text{home}) - N_p(\text{away})$. As expected, larger differences (+/-) often lead to (home/away) wins, but the mid-zone is highly unpredictable.

4 Advanced method

In all the following sections, unless mentioned otherwise, all the scores reported were obtained on our validation dataset, which consists of 66% of the challenge training dataset. We experimented with different proportions to calibrate the size of the validation data, and we found this proportion to be the most consistent. Another motivation is that we aim to approach the

best possible score on the real test dataset, which is two times larger than the entire training dataset, hence $34-66\%$. Class proportions are well stratified, and a random state was fixed to 10 for the validation sample to fairly compare all our models.

To predict the match outcome, we calculate the difference between the data of the home team and the data of the away team, and we use this new dataset. For the target, we denote -1 when the away team wins, 1 when the home team wins, and 0 in case of a draw.

4.1 Generalization of the number of points approach using Logistic Regression

We attempted to manually vary the coefficient for the points calculation in our baseline method to quantify the effect on prediction accuracy. After several attempts where we obtained an accuracy ranging between $40-42\%$, we came up with the idea of automating this process. This led us to the logistic regression algorithm. Indeed, what we are trying to do in our baseline is a kind of logistic regression in the sense that we learn a linear combination z of explanatory features, and then we use an indicator function (for example, $\mathbf{1}_{\{z>0\}}$ for home team wins) on this combination to predict the match outcome. Logistic regression is therefore suitable for automating this process by providing a learning framework where the weights of the linear combination are optimized to maximize the likelihood of the prediction, calculated by applying a sigmoid activation function on z to obtain prediction probabilities.

By applying logistic regression to the WDL features (sum of the last five matches), we achieve an accuracy of 45%, both on the training and validation splits. This is 3% higher than all our attempts to optimize the weights manually.

Sometimes, the recent dynamics of a team are not sufficient to determine if it will win its next match. A top team with a good overall dynamic throughout the season but with occasional periods of decline may not be well taken into account by only looking at the WDL data of the last five matches. To model this case, we added the WDL features for the entire season into the logistic regression model. This results in a gain of 3% in the model accuracy, increasing from 45% to 47.7169% and 48.0295% for the training/validation splits, respectively.

4.1.1 Error Analysis and Limitations

In an attempt to comprehend the model's errors, Figure 3 presents the confusion matrix contrasting the true labels with the predictions.

One observation immediately stands out: the model is incapable of detecting a draw match. Figure 4 illustrates the proportions of classes within the entire training dataset of the challenge. It is noteworthy that we have maintained these proportions in our **training/dev** split by specifying the **stratify** parameter in the **train_test_split** function of **scikit-learn**.

Next, we will explore other team features available in the dataset that may impact the match outcome.

4.2 Exploration of Other Features

Team statistical features can be categorized into two groups: offensive features and defensive features, each contributing in distinct ways to match outcomes. To win a match, a team needs to score more goals than its opponent, prompting consideration of features that could potentially have a significant impact on the match result. Conversely, to win a match, a team must concede

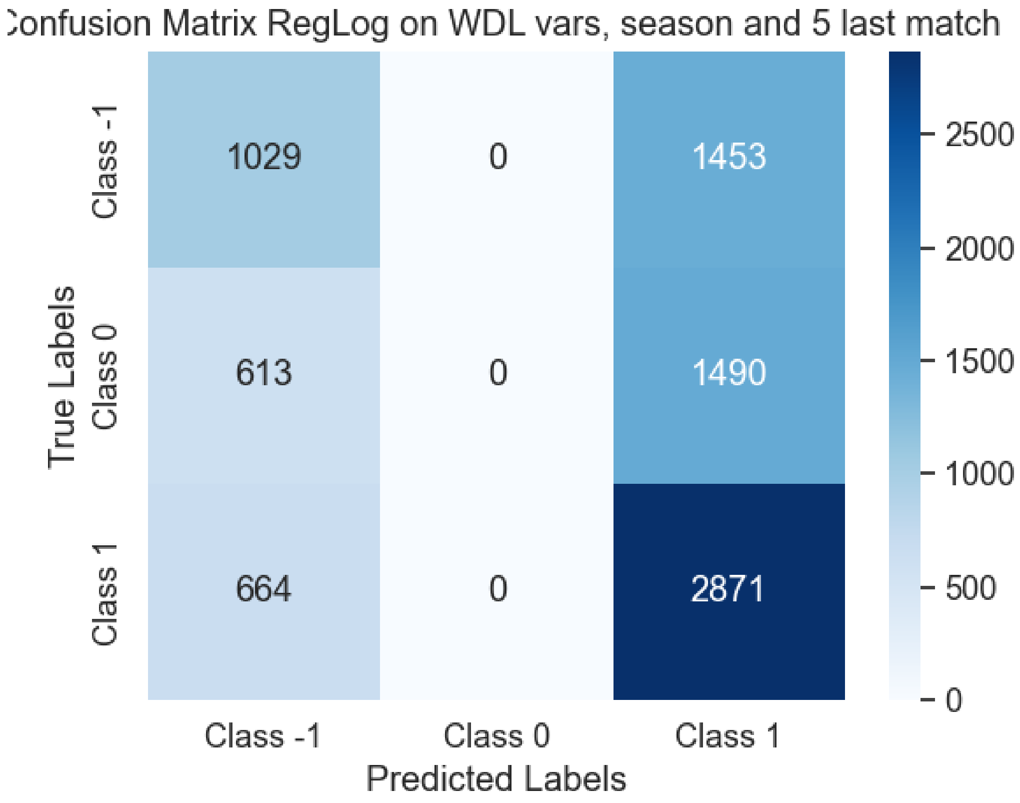

Figure 3: Confusion Matrix - Logistic Regression prediction using WDL features, season, and the last 5 matches; class -1 represents away team wins, class 1 represents home team wins, and class 0 represents draws; the model fails to detect a draw match.

fewer goals than its opponent, highlighting the importance of defensive features alongside offensive ones. Open play features such as passes, ball possession, dribbles, duels, etc., although their impact may be less direct than offensive and defensive features, can also influence match outcomes.

4.2.1 Prediction using Offensive Features

Offensive features have a considerable impact on the outcome of a football match. Building on the notion that these features collectively contribute to scoring goals and thus winning matches, we considered combining them to predict match results. To achieve this, similar to the approach taken with WDL features, we employ logistic regression on the features listed in Table **??**. We exclude **sum** formats as they are identical to **average** formats, differing only by a multiplicative coefficient, and thus highly correlated.

We train a Logistic regression on the offensives features in Table 2.

The model slightly outperforms the one on WDL features by reaching **48.9839%** and **48.4236%** respectively on the training and the validation split.

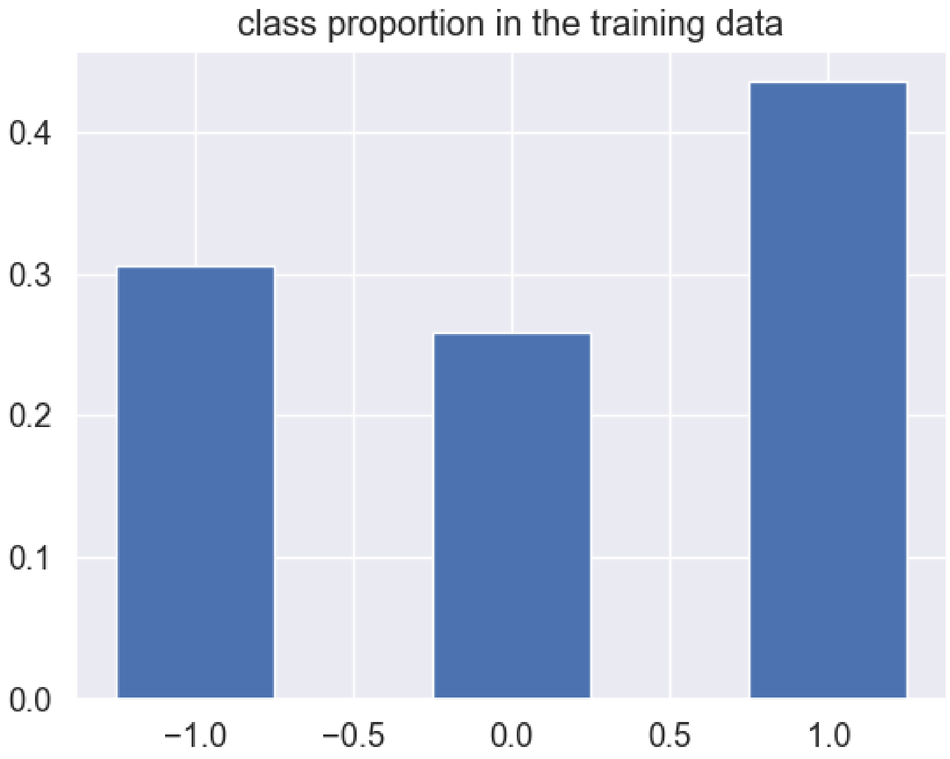

Figure 4: Class Proportion in the Training Data

4.2.2 Prediction using defensive features

Using the same idea as for offensives features, we train a logistic regression the defensives features. The features we chose are listed in the table 3:

The model significantly underperforms the one trained on offensive features by -3% in accuracy, achieving **46.7128%** and **45.4064%** on the training and validation splits, respectively. This result is quite expected as defensive performances typically have less impact than offensive performances in most football matches. To enhance the model's knowledge, we add open play features (see Table **??**) to defensive features and retrain the model. We observe an increase in performance, reaching **48.7688%** and **47.2536%** on the training and validation splits, respectively. It's noteworthy that adding open play features to offensive features induces a drop of -1% in performance.

4.3 Limitations of All Models

Similar to our approach in Section **4.1.1**, we plot the confusion matrix of the models presented in Sections **4.2.1** and **4.2.2**, noting the consistent challenge of detecting draw matches (see Figure 5). All our models exhibit nearly 0% accuracy in predicting draw outcomes. While this poses a significant challenge, it's not surprising considering the limited representation of draw outcomes in the data. Moreover, it aligns with the general consensus among football enthusiasts that predicting draw matches is inherently more difficult than predicting winners.

In order to understand why the models might be biased towards classes 1 and -1 or simply fail

Table 2: Offensives features, season (average and std), 5 last matchs (average and std)

Offensives vars
Team Shots Total
Team Shots Insidebox
Team Shots On Target
Team Attacks
Team Penalties
Team Dangerous Attacks
Team Corners
Team Goals

Table 3: Defensive (def) and open play (op) features, season (average and std), 5 last matchs (average and std)

Defensive vars
Team Saves (def)
Team Fouls (def)
Team Yellowcards (def)
Team Redcards (def)
Team Offsides (def)
Team Ball Safe (def)
Teams Injuries (def)
Teams Passes (op)
Team Successful Passes (op)
Team substitutions (op)
Team Ball Possession (op)

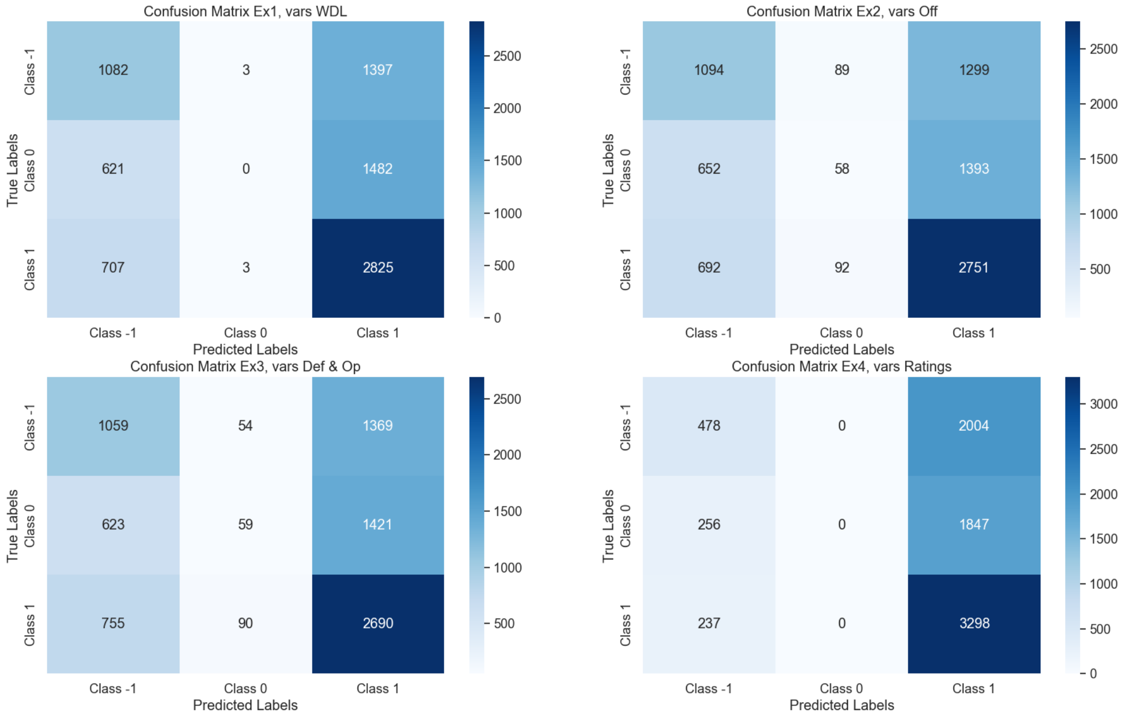

Figure 5: The confusion matrix for the four experts shows that those trained on WDL features and ratings are unable to detect the draw class. While the experts trained on offensive and defensive features can somewhat detect draw matches, it is still far from sufficient.

to detect class 0 because it is undetectable, we tried two approaches: **1.** training the models with the same number of samples for all 3 classes, and **2.** examining the data by applying Principal Component Analysis (PCA) to reduce dimensionality.

4.3.1 Train Models on Balanced Dataset

We re-split the entire training dataset, ensuring an equal number of samples for each of the three classes in the training split. To achieve this, we maintain the same training split as in the previous experiment, and we choose as many samples for the two other classes as there are in the minimal class. All remaining samples are allocated to the test split. We observed significant drops in test performance by doing so: **44.6906%** and **45.2396%** on the training/validation split for the WDL features, **42.9055%** and **46.0569%** for the offensive features, and **45.9833%** and **42.7766%** for the defensive + open play features. However, upon inspecting the confusion matrices of these models (see Figure **??**), we observe that this time the models more frequently detect class 0, albeit at the expense of fewer errors on the other classes.

The fact that the classes are imbalanced introduces a significant bias into the model. Since there is no priority class, and our performance metric remains accuracy, it is not a good idea to train the models on balanced classes.

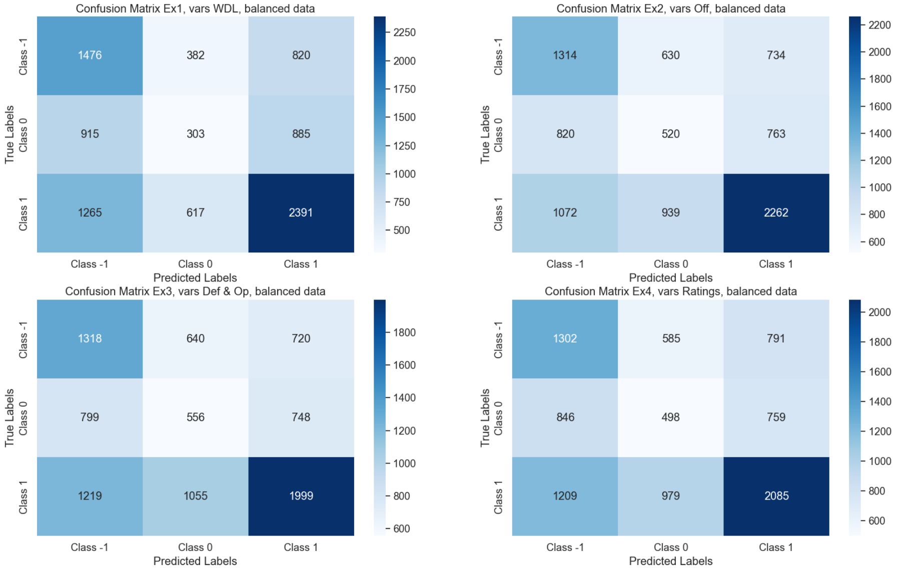

Figure 6: Confusion matrix for the 4 experts trained on balanced data, experts trained on WDL vars and Ratings are not able to detect the draw class, experts trained on off and def detect a little bit draw matchs but it's still far from sufficient.

4.3.2 Principal Components Analysis and Features Visualization

To understand why the model struggles so much to discriminate between samples, often predicting the majority class to achieve the highest accuracy, we attempted to observe certain features by applying Principal Component Analysis (PCA) to the WDL features. Indeed, the model trained solely on these six features performs similarly to the one trained on all 32 offensive features (the 8 from Table **??** following the 4 formats). By applying PCA with a dimension of **2** to all training data (no split), we capture over **80%** of explained variance, indicating that the visualization fairly represents the arrangement of features in the **6**-dimensional space. Figure **??** illustrates the arrangement of matches in reduced dimensions.

By observing this plot, we understand why the model fails to detect class 0 at all. The red points are indeed sandwiched between the black and green points, and unless a feature engineering method is applied to separate the positions or more discriminative features are found, it is impossible to classify these matches accurately with this data.

The same observation is made when we examine the offensive features on one hand, and the defensive and open play features on the other hand. We do not include the figures to avoid overburdening the report.

4.3.3 Data Transformation to Enhance Feature Discrimination

Upon examining the features, we considered making transformations on the data to make the features more discriminative. The idea led us to test a kernel method, and we naturally thought of the Support Vector Classification (SVC) algorithm.

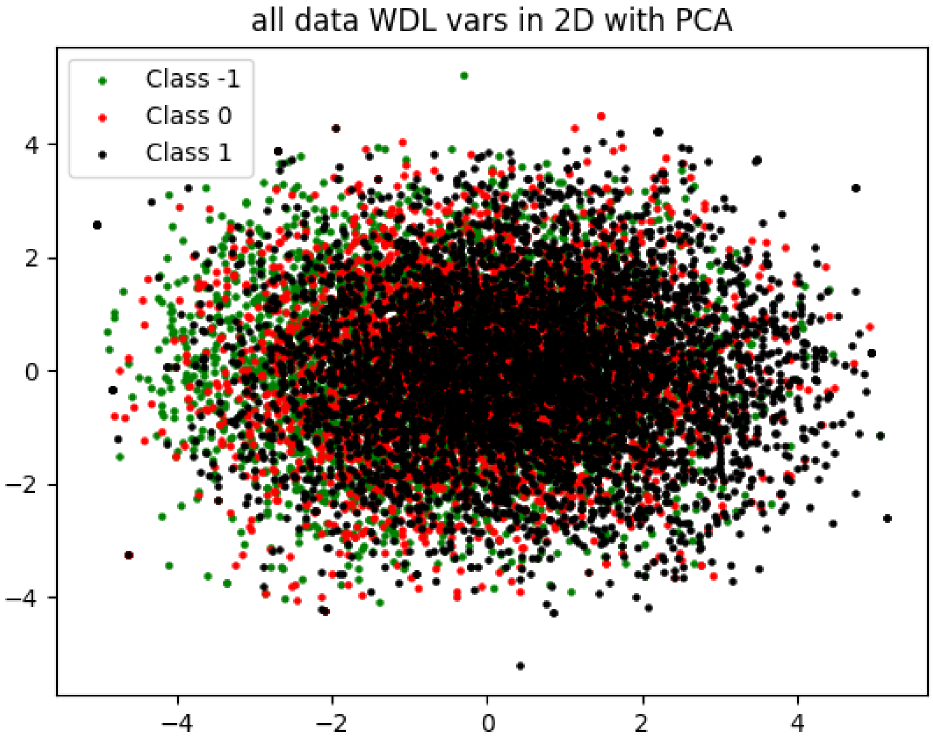

Figure 7: Match points observed in a 2D space, with 80% of the variance explained. The visualization confirms that the features are not sufficiently discriminative, resulting in models that always predict the majority class to maximize their accuracy.

We performed a grid search on the training data, with the WDL features reduced to three dimensions by PCA, retaining **90%** of the variance. We kept our validation set intact to calculate the final score of the best model. The results are documented in Table 4. We utilized KFold cross-validation with 5 folds.

Table 4: grid search parameters for the SVC algorithm

Params name (as in sklearn)	Values
C	[0.1, 1, 10, 100]
gamma	[0.1, 0.01, 0.001]
kernel	[rbf, linear, poly, sigmoid]
best model	C: 10, gamma: 0.1, kernel: rbf
validation score	**47.7586%**

The best model obtained does not perform better than logistic regression trained on the same 3D data, which has a validation score of **47.9310%**. Therefore, the conclusion remains the same: the features are not sufficiently discriminative.

4.4 Search for More Discriminative Variables in the Players Dataset

Most player variables have equivalents for teams (goals, shots, saves, interceptions, etc.), and our goal is to predict the team's result rather than each player's. Also, we know that the team with the best players is not necessarily the one that wins, often quite the opposite. Paradoxically, it doesn't make sense to say that the team winning a football match has the best players at each position, so trying to predict the match result by comparing individual players is highly limited. This poses a very constrained framework for approaching player data to predict the match winner, as few individual variables can directly impact the outcome and predict the match winner. One of these rare variables is the **RATING**. Indeed, it measures the individual and collective performance of each player assigned to each match; it is calculated from several player performance stats, the match outcome, their involvement in their team's play, etc., making it a potentially decisive variable for predicting the winner of a future match given the ratings from previous matches.

For ratings, we have four variables: Player Rating season average, Player Rating season std, Player Rating 5 last match average, and Player Rating 5 last match std. It is necessary to synthesize the values of the players on a team into a single variable reflecting the team's performance during the match. One approach is to simply average the values for all players on each team. Training a logistic regression model on the data obtained for the teams after splitting the samples in the same way as in all previous experiments yields scores of **43.4159%** and **44.9817%** on the training and validation splits, respectively. These performances are far from those obtained for the three previous models. We thought that averaging the ratings of all players to get team ratings diminishes the predictive power of these features; after all, only 10 outfield players and 1 goalkeeper are on the field per match on average, whereas there are 19 players per team per match in the dataset on average. To address this, we used the variable Player Minutes Played 5 last match average. The idea is that generally, the most performing players and those most likely to be fielded for the next match are those who have played the most in recent matches. Thus, we calculate team ratings by taking the 10 players who have played the most in the last 5 matches, in addition to the goalkeeper, and average the ratings of these 11 players by the number of minutes played. Training a logistic regression model on the new features thus created significantly improves performance, increasing to **46.3064%** and **46.6995%** on the training and validation splits, respectively. Figures 5 and 6 show the confusion matrices obtained when predicting with the rating variables while training on balanced and unbalanced data. The model faces the same issue as the previous three, with class 0 being undetectable. Performing a 2D PCA on the 4 rating variables, retaining **83%** of variance and observing the match points, yields the plot in Figure **??** on which we can observe again class 0 sandwiched between classes -1 and 1.

4.4.1 Another approach: Using the goal diff target to explain the difficulty in detecting a draw match

The conclusion of our analyses is that the features present in the dataset are highly non-discriminative. Indeed, as we can see from the different confusion matrices and PCA plots, the models tend to favor the majority class, namely 1, in order to maximize their accuracy, and ignore draw matches (0) due to the lack of means to distinguish them. This can also be explained by looking at the distribution of goal differences. We observe that two-thirds of the matches have a goal difference in {-1, 0, 1}, and among these, a small proportion represents draw matches compared to wins (home or away); thus when there is a draw match, it would naively be assimilated by a model as either a loss or a win, which would justify the observed results. In summary, since most wins are obtained with a one-goal difference, a model would struggle to distinguish draw

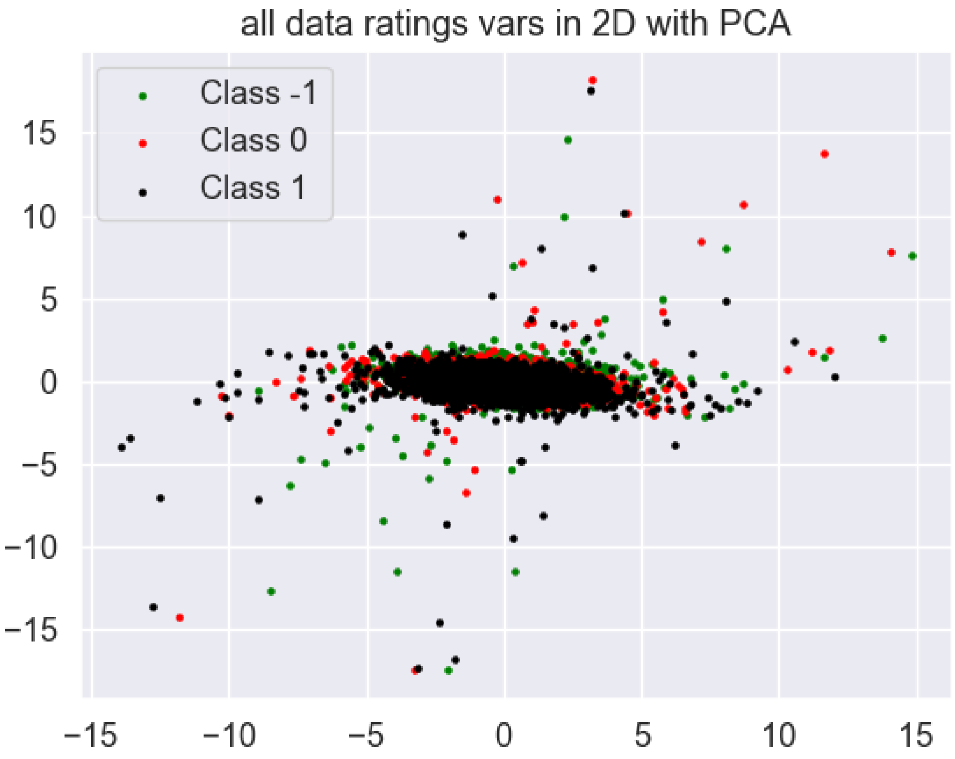

Figure 8: Matchs points observed in a 2D space, 83% of variance explained. The visualization confirms that the ratings features aren't also enough discriminative, resulting also in a model that always predict the majority class to miximize his accuracy.

matches and would opt in favor of a win in case of doubt because this class is dominant.

5 Final Model: Expert Aggregation

In the end, we have four models trained on different features for the same samples, with pairwise close accuracies. The four models remain highly biased towards classes 1 and -1 and fail to predict class 0, given the lack of discriminative power in the features. Nevertheless, we would like to leverage the four models, and thus, inspired by the Random Forest algorithm, we considered creating a mixture of experts (MoE) from the four models. Given the predictions of the four experts, the prediction rule of the MoE is majority voting. In case of a tie in votes, the model or group of models with the best average accuracy prevails. By applying this technique, we achieved an accuracy of **49.0517** on our validation dataset, representing a gain of over **0.5%** compared to the best expert. This is a significant performance improvement.

6 Results on Public and Private Datasets

We submitted the MoE prediction on the true test dataset of the challenge after training each expert on all available training data. Consequently, we obtained a score of **48.71%** on the public

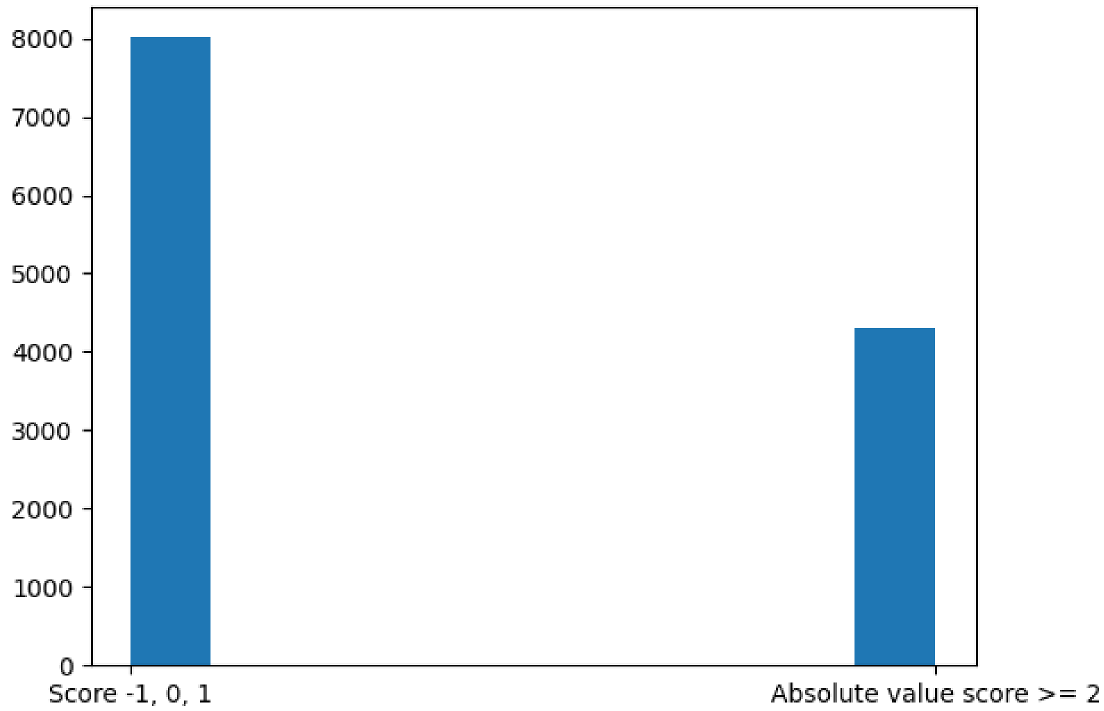

Figure 9: Histogram of the goal difference target

leaderboard and **48.47%** on the private leaderboard.

7 Future Work

Here are some ideas that we didn't have enough time to explore and will be the subject of future work on this challenge:

- Feature engineering: design better informative features for example Elo rating, overperformance features.
- Combine balanced and unbalanced models in a MoE algorithm.
- Utilize deep learning models like attention mechanisms on players' raw counts data to create match-contextualized players and teams embeddings. Such a model should be pre-trained on a simpler task before fine-tuning it on match winner prediction. We still need to identify suitable pretraining tasks.

References

1. Predictive analysis and modelling football results using machine learning approach for English Premier League. Available online: https://www.researchgate.net/publication/324072605_Predictive_analysis_and_modelling_football_results_using_machine_learning_approach_for_English_Premier_League.
2. Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*. doi:10.1007/s13748-016-0094-0.

3. Evaluating Soccer Match Prediction Models: A Deep Learning Approach and Feature Optimization for Gradient-Boosted Trees https://arxiv.org/pdf/2309.14807.pdf.