Predicting, Understanding and Betting the Soccer Match

Team 14: Xin Bing, Henry Tang, Andrew Yuan, Yuxuan Zhao

March 2021

1 Problem Statement and Main Findings

Our team aims to build statistical models: (1) to predict a match's outcome, namely "home team wins", "draw" or "away team wins"; (2) to find out what mix of components of players skills and which team attributes are significant for predicting the match result; (3) to construct a new betting strategy for making profits. We summarize our major findings as follows.

- **Prediction:** The three-class classification accuracy of our proposed model is around 52%, 20% higher than the random guessing. We find that our classifier has a higher accuracy (around 70%) of predicting "home team wins" and "away team wins" but fails to capture "draw". We also find that using more complex models such as deep neural networks does not improve the accuracy and the phenomenon of misclassifying "draw" remains.
- **Interpretation on team attributes and player skills:**
  - Given the players' attributes, all team attributes are not significant to predict the match outcome, with the exception given to "chanceCreationPositioningClass" of the home team, an attribute mainly affecting the overall defense of the home team. The chance of winning a match is lower if this team attribute is equal to Free (comparing to Organised).
  - Player's attributes are summarized by 4 latent factors, which could be interpreted as *offensive ability*, *midfield ability*, *defensive ability*, and *goal keeping ability*. Each factor is only strongly associated with a subset of original features.
  - We cluster all players to 4 (non-overlapping) groups which represent different positions including forward, midfielder, defender and goal keeper. This partitions the 11 players of each team into 4 positions.
  - For players in each position, we select the significant latent factors for predicting the match outcome. Our results show that for any team, the offensive ability of its forwards, midfielders and defenders is the most significant factor for winning a game in a positive way, while the defensive ability of the midfielders and defenders of its opponents is also significant, but in a negative way. For players as the goal keeper, the only significant ability is its goal keeping.
  - Our results also offer insights for constructing a team to improve its winning rate. Finally, our results indicate the home advantage as well.
- **Betting:** We construct a new betting strategy which has an overall 5% profits. Motivated by our prediction findings, our betting strategy is based on estimating the *conditional probability* of a home team win conditioning on that this match is either a home team win or an away team win and estimate a draw purely by empirical estimates of the training set. Such strategy outperforms the strategy that jointly classifies three outcomes together.

The rest of this report is organized as follows. We first show in section 2 how to represent the overall strength of a team thoroughly from multiple raw datasets. Our representation reflects the evaluation of a team from multiple aspects, including the offensive ability, midfield ability, defensive ability and goal keeping ability of its forwards, midfielders, defenders and goal keepers. Then, we show in section 3 how to use our team representation to predict the match outcome and how well the prediction is. At last, we show in section 4 the performance of several betting strategies constructed based on the predicted match outcome distribution.

2 Data Cleaning and Feature Engineering

As our goal is to classify the outcome of each match, we treat each match (in the match dataset) as one data point with the categorical response Y defined as

$$Y = \begin{cases} \text{H}, & \text{if home goals are greater than away goals;} \\ \text{D}, & \text{if home goals are equal to away goals;} \\ \text{A}, & \text{if home goals are less than away goals.} \end{cases} \tag{2.1}$$

We note that the level of Y depends on the difference between home goals and away goals, hence we further treat Y as ordinal, that is, the order of three levels of Y is also meaningful. We focus on using features that can potentially represent the overall strength of a team in order to predict Y. Specifically, we use the following data to represent the overall strength of a team at the time of a particular match: (1) the team attributes for both the home team and the away team; (2) the player attributes for each of the 11 players from the home team and each of the 11 players from the away team. To this end, we first conduct an initial data manipulation that finds all related features for each match from the provided datasets, as described below. In Section 2.1, we further process the features by additional screening and adjustment. The structure of thus processed features motivates us to consider a bi-clustering scheme in Section 2.2 to represent the features in a more compact and interpretable way.

Data Preprocessing The initial match dataset contains information of the teams and players in each match. For each match, we inserted the corresponding player attributes (from player_attributes.csv) for all 22 players and team attributes (from team_attributes.csv) for both teams in the match into the corresponding row. One small issue that arose was the fact that there are multiple rows corresponding to the same player in the player attributes database. To break ties, we chose the most recent entry before the date of the match. A similar procedure was done for the team attributes. If no such entry existed for either a team or a player, the corresponding attributes in that row of the merged database would be left as empty.

2.1 Initial Feature Manipulation

In view of the features we obtain after preprocessing, one natural idea is to use *all* of them as independent variables to classify the outcome. However, there is one caveat of simply concatenating all attributes of 22 players (11 home and 11 away) in each match. The players order may be meaningless for match result prediction. For example, the players may be listed in alphabetical order, but the same match prediction should be made regardless of the order. To solve that issue, we sort each attribute among the 11 players. In other words, instead of recording the player 1 rating, player 2 rating, ..., player 11 rating, we record the highest player rating, the 2nd highest player rating, ..., and the 11th highest player rating. Such a process will always output the same result for a match no matter what the order of the 11 players is. We decide to

discard three non-numerical features: preferred foot, attacking-work-rate and defensive-work-rate[1]. Eventually, for each team in a match, we collect 385 features from 11 players each with 35 numerical features.

For the team attributes that have both numerical values and categorical levels, we only keep the numerical values. Since "buildUpPlayDribblingClass" has too many missing values in the numerical column, we only keep its ordinal level. In the end, for each team, we retain 12 team attributes and 4 of them are ordinal.

For the original match dataset, we exclude the columns of "stage", "country id", "league id", "season" and "date". A more sophisticated modelling approach could treat these factors as random effects or model them separately. See Section 3 for a brief discussion. Betting odds are also excluded because we find that although they are strong predictors of the outcomes, they do not provide additional prediction power given our collected team and player attributes.

After the above adjustment, for each match, we have 794 features in total with 24 team attributes and 770 player attributes. Then we select all the match data with complete features for further modeling step, which yields 16686 matches. Since the number of complete data points is already pretty large, we do not create more complete rows by filling up the missing entries to avoid introducing more noise. Overall, we have $n = 16686$ matches and for each $i = 1, \ldots, n$, our data consists of (Y_i, X_i), $1 \leq i \leq n$, with the response Y_i defined in (2.1) and the feature vector $X_i \in \mathbb{R}^p$ with $p = 794$.

We found that (1) the (Spearman's rank) correlation among all (continuous) features exhibit strong block structure; (2) there are many features (especially the same attribute among different players from the same team) that are strongly correlated. Figure 1 below illustrates these two points via the correlation matrix of a subset of features, including the "overall", "potential", "finishing" and "crossing" attributes of 11 players for both home team and away team. This strong block-wise structure of the correlation matrix suggests to further conduct dimension reduction of the current features to alleviate the multi-collinearity, as described in Section 2.2.

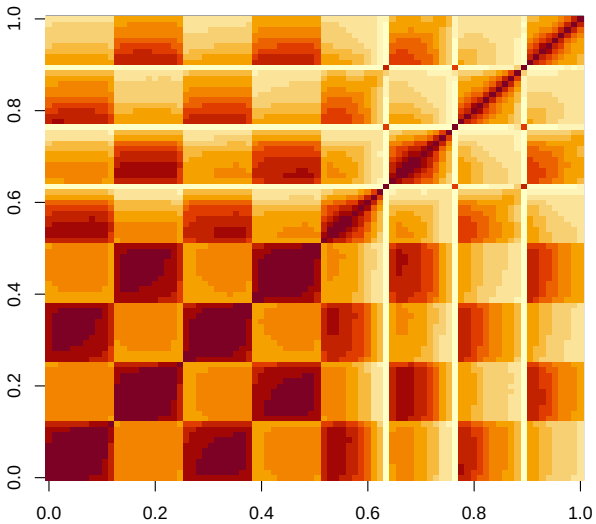


Figure 1: Correlation matrix of a subset of features. Darker squares are more correlated. The first 2 blocks represent the "overall" attributes of 11 players for home team and away team. The rest blocks follow the order of the "potential", "finishing" and "crossing" attributes.

[1]We found the data of attacking-work-rate seems weird as most of the goal keepers could have the medium level.

2.2 Feature extraction via bi-clustering scheme

Figure 1 suggests two structures of the attributes of different players:

- different attributes of the same player are correlated;
- different players of the same team could share the same pattern of attributes.

This motivates us to use a bi-clustering scheme to reduce both the number of attributes representing each player and the number of attributes representing all players in each team.

Intuitively, we want to categorize the 11 players of each team via their positions, namely, forward, midfielder, defender and goal keeper. By averaging the attributes of players in each position, we can come up with the overall player attributes of each position.

On the other hand, we believe that the 35 attributes of each player are generated from several latent factors that measure certain representative abilities of the players. We define *representative abilities* as attributes that could be generated from four latent factors, including the offensive ability, the defensive ability, the midfield ability and the goal keeping ability. We thus propose to represent the attributes in the factor level by using either dimension reduction techniques or factor model analysis.

The above procedures can be described succinctly via a bi-clustering modelling procedure. Let $\mathbf{Z} \in \mathbb{R}^{n\times d}$ denote the data matrix consisting of n players and their d attributes for a particular match. We consider the following model

$$\mathbf{Z} = A\mathbf{C}B^\top + \mathbf{E} \tag{2.2}$$

where $A \in \{0,1\}^{n\times c}$ is the membership matrix (each row has only one non-zero entry as 1), $B \in \mathbb{R}^{d\times r}$ is the attribute loading matrix and $\mathbf{E}$ is some additive noise with mean zero. More intuitively, the matrix A assigns n players to c non-overlapping clusters. Each row of B represents one attribute which is generated from a linear combination of the r latent factors. We choose to fix $c = 4$ in the hope that we can find 4 non-overlapping clusters corresponding to the forwards, the midfielders, the defenders and the goal keeper. Then the matrix $\mathbf{C} \in \mathbb{R}^{4\times r}$ represents the r latent factors of four positions. For instance, if the first cluster corresponds to the forward position and the first latent factor represents the offensive ability, then $\mathbf{C}_{11}$ means the overall offensive ability of a forward player.

Our goal is to recover A and B, hence to further recover the summarized attributes matrix $\mathbf{C}$. Then for each player, we can use only the r latent attributes from $\mathbf{C}$ to summarize this player's r representative abilities. For each team in each match, we will average the representative abilities of 11 players for each position and construct $4 \times r$ position-specific representative abilities. For instance, if we have $r = 3$ latent factors, representing the offensive ability, the defensive ability and the goal keeping ability, then our new feature set for each team in each match contains the following $4 \times r = 12$ items:

forward offensive ability	forward defensive ability	forward goal keeping ability
midfielder offensive ability	midfielder defensive ability	midfielder goal keeping ability
defender offensive ability	defender defensive ability	defender goal keeping ability
goal keeper offensive ability	goal keeper defensive ability	goal keeper goal keeping ability

By using this strategy, we construct a new feature matrix $\tilde{\mathbf{X}}$, in which each row represents one match and the columns of $\tilde{\mathbf{X}}$ represents the new features, including the attributes of both home and away teams and the $4r$ position-specific representative abilities.

Estimation of A and B: Since neither the position of each player (or, equivalently, A) nor the attribute loading matrix (B) is known, we need to both cluster different players into four clusters (find A) and estimate the loading matrix B.

Estimating A is equivalent to to cluster players into four positions. To this end, we use the k-means algorithm[2] with the number of clusters specified to 4. From our clustering results, we notice that it is the easiest to identify goal keepers from the given 35 attributes whereas distinguishing midfielders from either forwards or defenders is the most difficult. While it is a bit harder to distinguish other three locations, there is still clear boundary. Figure 2 illustrates our resulting clusters of a subset of players on the first two principal components (PC) of the 35 attributes. There is around 13% of matches that the clustering gives improper result: two goal keepers in the same team or no player for some location. We exclude those matches and there are still 14497 matches left.

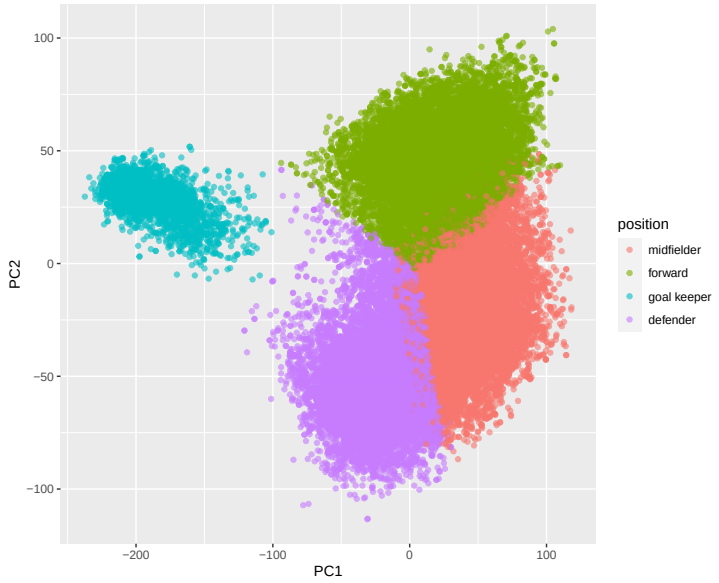


Figure 2: Each dot represents a player and the axes represent the first two PCs of 38 attributes

For estimating B, we consider two options. The first method uses the Principal Component Analysis (PCA) to estimate B. Let $U \in \mathbb{R}^{p\times r}$ be the right singular vectors of $\mathbf{Z}$ corresponding to the largest r singular values. One natural choice of estimating B is to use U and then recover $A\mathbf{C} \in \mathbb{R}^{n\times r}$ by $\mathbf{Z}U \in \mathbb{R}^{n\times r}$. We select the number of latent factors, $r = 4$, based on the following scree plot. However, U is fully dense and it is hard to interpret the resulting factors, i.e. the columns of $\mathbf{Z}U$. Indeed, the kth column of $\mathbf{Z}U$ is a linear combination of the columns of $\mathbf{Z}$, as

$$[\mathbf{Z}U]_{\cdot k} = \sum_{j=1}^{p} U_{jk}\mathbf{Z}_{\cdot j}.$$

Since we only know what $\mathbf{Z}$ means, it is easy to interpret the meaning of $[\mathbf{Z}U]_{\cdot k}$ if $U_{\cdot k}$ contains only *a few* large entries (or spiked). We thus rotate U to obtain $\tilde{U}$ by using the varimax rotation[3] (which, intuitively speaking, finds the rotation matrix Q such that $\tilde{U} = UQ$ is as spiked as possible). Applying to the dataset, we record the top 10 attributes (ordered by the largest coefficients in absolute values) for each column of $\tilde{U}$ in Table 1 below and use them to

[2] https://en.wikipedia.org/wiki/K-means_clustering
[3] https://en.wikipedia.org/wiki/Varimax_rotation

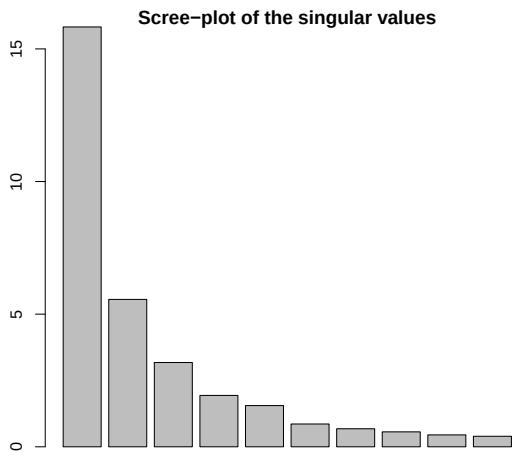


Figure 3: Magnitudes of each singular value of $\mathbf{Z}$ in a decreasing order. The x-axis represents the order of singular values.

name the four latent factors. Generally speaking, the four factors represent offensive ability, defensive ability, goal keeping ability, skills & athleticism.

factors	top 10 attributes					interpretation
factor 1	finishing	volleys	penalties	long shots	shot power	offensive ability
	positioning	free kick accuracy	vision	ball control	curve	
factor 2	sliding tackle	marking	standing tackle	interceptions	aggression	defensive ability
	stamina	long passing	jumping	finishing	overall rating	
factor 3	overall rating	gk positioning	gk handling	gk reflexes	potential	goal keeping ability
	gk kicking	gk diving	reactions	jumping	heading accuracy	
factor 4	strength	balance	agility	acceleration	sprint speed	skills & athleticism
	heading accuracy	crossing	aggression	dribbling	stamina	

Table 1: Top 10 attributes of each latent factor. The attributes are ordered according to the largest entries of $\tilde{U}$ in absolute values.

Although $\tilde{U}$ is spiked, we found that it is still difficult to interpret the 3rd and 4th factors since these columns of $\tilde{U}$ contain large coefficients for attributes corresponding to disparate abilities (for instance, gk-related attributes vs heading accuracy). Thus, we consider an alternative way of estimating B by using the procedure proposed in [2]. The procedure in [2] also provides a way of selecting r. In this dataset, it selects $r = 4$. The advantage of this resulting estimator $\hat{B}$ is that, for each column of $\hat{B}$, there exists some rows of $\hat{B}$ that *only* have non-zero entries in this column. In another words, for each latent factor, there exists some attributes (which we denote *pure attributes*) that are *only* associated with this latent factor. This nice feature allows us to easily interpret the meaning of the resulting factors from their corresponding pure attributes. The following Table 2 summarizes the pure attributes (bold fonts) for each latent factor as well as the top 10 attributes according to columns of $\hat{B}$. It is clear that using $\hat{B}$ yields more interpretable latent factors. Finally, the matrix $A\mathbf{C}$ is estimated by $\mathbf{Z}\hat{B}(\hat{B}^\top\hat{B})^{-1}$.

Summarizing, we propose two ways of estimating B: PCA and the procedure in [2]. Both of them yields $r = 4$ latent factors with interpretations given in Table 1 and Table 2, respectively. In our later modelling, we will only present the result by using $\hat{B}$ from [2] due to its interpretability advantage.

In this end, our new feature $\tilde{\mathbf{X}}$ contains 14497 matches with 56 features, greatly reduced from the original 794 features.

factors	top 10 attributes					interpretation
factor 1	**finishing**	**volleys**	positioning	shot power	penalties	offensive ability
	long shots	dribbling	reactions	overall rating	heading accuracy	
factor 2	**curve**	**free kick accuracy**	crossing	long passing	agility	mid-fielder ability
	vision	balance	short passing	dribbling	strength	
factor 3	**standing tackle**	**sliding tackle**	marking	interceptions	aggression	defensive ability
	stamina	long passing	short passing	reactions	overall rating	
factor 4	**gk diving**	**gk reflexes**	gk handling	gk positioning	gk kicking	goal keeping ability
	overall rating	reactions	potential	sprint speed	acceleration	

Table 2: Top 10 attributes of each latent factor with pure attributes in bold fonts. The attributes are ordered according to the largest entries of $\hat{B}$ in absolute values.

3 Match Outcome Prediction

Our prediction exploration consists of two parts: on the reduced feature matrix from section 2.2 and on the complete feature matrix from section 2.1. We show that multinomial logistic regression on the reduced dataset yields the same accuracy level compared to multinomial logistic regression, random forest and nueral network classifier on the complete feature dataset.

We use $\mathbf{X}$ to denote the $n \times p$ feature matrix for either the reduced or the complete feature matrix. In our modelling, we treat the feature matrix $\mathbf{X}$ as a deterministic quantity and make the assumption that $\{Y_i|X_i\}_{i=1}^n$ are independent, namely, given the feature $\mathbf{X}$, the response Y_i is independent across $1 \leq i \leq n$. Although a preliminary analysis of checking the time dependence between matches (at least for the same team) is more appropriate, we do not have enough time to pursue this direction. In practice, in the presence of time dependence, one should use time series models to model this dependence and remove it before running into the regression step.

3.1 Prediction with Reduced Feature Matrix

Since the response has three levels, we propose to use multinomial logistic regression (MLR) to model (Y_i, X_i). Mathematically, MLR assumes

$$\log\left(\frac{\mathbb{P}\{Y_i = \mathrm{H}\}}{\mathbb{P}\{Y_i = \mathrm{A}\}}\right) = \beta_0 + \beta_1^\top X_i, \quad \log\left(\frac{\mathbb{P}\{Y_i = \mathrm{D}\}}{\mathbb{P}\{Y_i = \mathrm{A}\}}\right) = \alpha_0 + \alpha_1^\top X_i, \quad \forall 1 \leq i \leq n.$$

The coefficients $\alpha = (\alpha_0, \alpha_1)$ and $\beta = (\beta_0, \beta_1)$ can be estimated jointly by the Maximum Likelihood Estimator (MLE). The main reason why we choose MLR is mainly two folded: (a) We want to maintain the log-odds interpretability of the coefficients; (b) Minimizing the negative likelihood can be easily combined with penalization to select predictive features, as described below. We also show that more complex models *do not* lead to a better prediction accuracy in section 3.2.

In order to select significant features that are useful for predicting the match outcome, we propose to add the ℓ_1 regularization (penalty) of the coefficient vectors α_1 and β_1 when we minimize the negative of the log-likelihood function. The ℓ_1 regularization has been successful for both variable selection and prediction in many applications, see, for instance, [4, 3], just to name a couple. We use the glmnet in R to obtain the estimated α and β.

We now state our results using the extracted features from Section 2.2. Recall that for each match, in addition to the 24 team attributes (both home and away), we construct 4 latent features representing the *offensive, defensive, mid-fielder, goal keeping* abilities of *each position* for both the home team and the away team. This yields an additional 32 features for the players.

In total, we have 56 features before running the MLR. We consider two aspects: prediction and variable selection.

Prediction: We use the misclassification rate to quantify the prediction performance of the trained model. Specifically, we randomly split[4] the data into 70% training data and 30% test data. We train the model by running the MLR on the training set. The misclassification rate of our trained model validated on the test set is 48.2%.

By further exploring the confusion matrix in Table 3, we found that the level D is hard to capture. In another words, our classifier misses most of the D in the test set. This phenomenon remains for other more complex classifiers. See Section 3.2 for details.

Percentage		True label			
		A	D	H	Total
Predicted label	A	12.0	6.1	5.8	23.9
	D	0	0	0	0
	H	16.9	19.4	39.8	76.1
	Total	28.9	25.5	45.6	1

Table 3: Confusion matrix for the multinomial logistical regression with reduced features

Feature selection: As the ℓ_1 regularzation selects the predictive features, we have the following finding:

- The only significant team attribute is "home-chanceCreationPositioningClass" which has two levels: Organised and Free Form. Specifically, Free Form leads the log-odds of a home win to decrease by 0.0033, with the other features fixed. On the other hand, Free Form leads the log-odds of an away win to increase by 0.0035, with the other features fixed.

  Since all the other team attributes are not significant, we conclude that given the latent attributes constructed from the player attributes, only "home-chanceCreationPositioningClass" is significant for prediction.

- In terms of the four latent factors for different positions of both home team and away team, the following table summarizes the significant latent factors (the names of latent factors are from Table 2). We have the following observations.

  - For either team to win, the offensive ability of the team itself is important for all positions except the goal keeper. Another important factor is the defensive ability of the opponent. Therefore, our model suggests that, for either team to win, the most effective way is to enhance the offensive ability for all positions except the goal keeper. The attributes that are most important for measuring the offensive ability are summarized in Table 2 with exact coefficients can be found from $\hat{B}$.
  - For either team to win, the defensive ability of only the midfielder seems significant. The defensive ability of the home defender could be characterized by "home-chanceCreationPositioningClass", the only significant team attribute. Intuitively, if "home-chanceCreationPositioningClass" is Organized, the home defensive ability should be relatively stable as less players are out of position[5].

[4] To maintain the balance among H, D and A, we randomly sample data within each category.

[5] https://fifaforums.easports.com/en/discussion/268115/custom-tactics-build-up-organised-vs-free-form-investigation-finding-good-information

– The significant factors for home win and away win have the same pattern but are slightly different. For instance, the goal keeping ability of the goal keeper of the away team is significant for a home win whereas this is not the case for an away win.

– From the magnitudes of the non-zero coefficients, increasing the same amount of offensive ability for an away win is less effective than that for a home win. This indicates the home court advantage.

	Home win				Away win			
Positions	offensive	mid-fielder	defensive	goal keeping	offensive	mid-fielder	defensive	goal keeping
home forward	0.34	0.07	0	0	-0.08	-0.05	0	0
home midfielder	0.32	0.06	0.10	0	-0.01	0	-0.13	0
home defender	0.20	0	0	0	0	0	-0.29	0
home goal keeper	0	0	0	0	0	0	0	0
away forward	-0.05	-0.02	0	0	0.25	0.10	0	0
away midfielder	-0.24	0	-0.18	0	0.13	0	0.12	0
away defender	-0.03	0	-0.27	0	0.04	0	0	0
away goal keeper	0	0	0	-0.08	0	0	0	0

Table 4: Estimated coefficients of α_1 and β_1 for the 4 latent attributes of 4 positions of both home team and away team. The columns represent the 4 latent factors while the rows represent the 4 positions of both home and away team. The numbers are corresponding estimated coefficients.

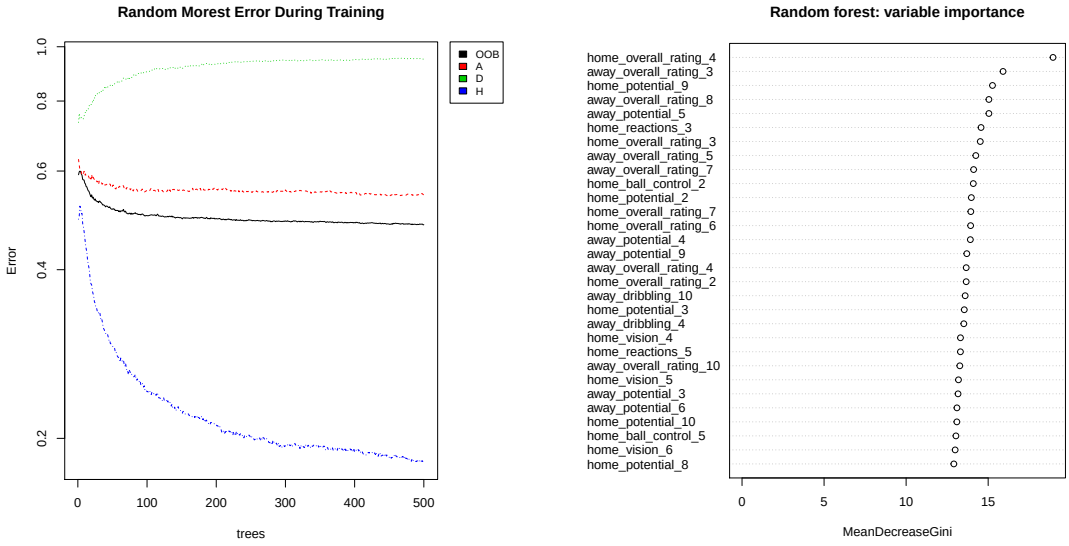
Remark 1 (Alternative modelling)**.** For variable selection, instead of using the ℓ_1 penalty, a more classical approach is to use the asymptotic properties of the MLE to construct hypothesis testing for α_1 and β_1. This can be done via, for instance, the Wald test. However, in order to correct for multiple testings, one needs more sophisticated approaches, such as controlling the false discovery rate. Another direction is to conduct the best subset selection when the feature dimension is not too large.

Although the levels of Y_i are in fact ordinal, MLR does not utilize this information. Ordinal logistical regression may be a more proper choice. To the best of our knowledge, we do not know any any robust implementation of ordinal logit model which allows ℓ_1 penalty for variable selection. If one considers the probit regression, we refer to [1] for the study of the ordered probit regression.

3.2 Prediction with Complete Feature Matrix

It may be the case that more complex models could improve the classification accuracy and benefit from using more features. Thus in addition to the MLR, we also trained two more complex classifiers: random forest and deep neural networks, on the complete feature matrix from section 2.1. Both random forest and deep neural networks (via stochastic gradient descent) are known to prevent overfitting to the data based on empirical evidence. In summary, the overall outcome prediction accuracy from random forest and neural network does not improve upon the MLR. The trained classifiers fail to predict D. This indicates that the data set does not contain enough features for the classification task. More data containing orthogonal information to current datasets are needed for further improvement.

Multinomial Logistic Regression We run the same MLR model now with the complete feature matrix. The misclassification rate is around 47.9%, comparable to what we had in Section 3.1. However, the selected significant features are not informative due to the strong



(a) (b)

Figure 4: Random forest results. (a) Out-of-bag error (b) Variable importance

block-wise correlation as mentioned in Section 2.1. More specifically, the ℓ_1 regularized estimator in general only selects (in almost arbitrary way) one of the predictors that are highly correlated.

Random Forest The out-of-bag error metric evaluation is a good tool to evaluate how well the random forest generalizes during the training process. From fig. 4 and table 3, we see the prediction of H is the most reliable, as its accuracy increases as the model gets more complex (more trees in the forest). In contrast, the prediction of D is the least reliable and its prediction even gets worse for more complex model. The result here again indicates the input features do not have much prediction power on D. The returned variable importance suggests the fourth best player of the home team is the most relevant factor to the outcome prediction, which does not make intuitive sense. Again, that may be due to the collinearity among the players' attributes, as we mentioned in section 2.1.

Neural Network We use a neural network to train a classifier that classifies each match as a home win, away win or draw. The neural network has 5 fully connected layers and we use the cross entropy loss and Adam optimizer to train the classifier. The layers have output sizes of 512, 256, 128, 32, and 3. We train for 1000 epochs with learning rate 0.01. The accuracy on both the train and test set is 51.3%. These numbers are comparable to the accuracy using the multinomial logistic regression and shows that simply using a more complex model does not yield better accuracy. Similar to the confusion matrix of MLR in table 3, the confusion matrix table 5 shows our neural network model is not able to correctly classify a match as a Draw often while it is best at predicting Home Wins.

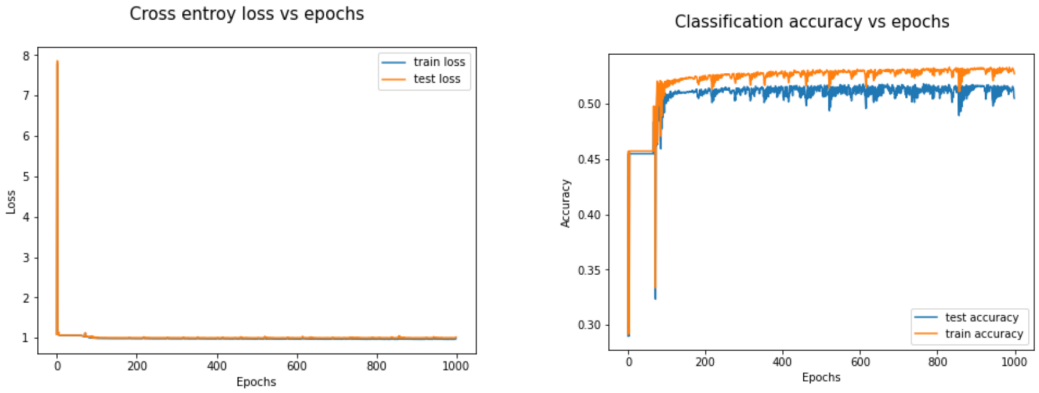


Figure 5: Training loss and accuracy of neural network classifier

Percentage		True label			
		A	D	H	Total
Random Forest Predicted label	A	13.2	7.2	7.2	27.6
	D	1.0	0.8	1.3	3.1
	H	14.9	17.2	37.2	69.3
	Total	29.1	25.2	45.7	100
Neural Network Predicted label	A	11.4	5.5	5.5	22.4
	D	0.3	0.3	0.1	0.7
	H	17.4	19.7	39.8	76.9
	Total	29.1	25.5	45.4	100

Table 5: Confusion matrix of the trained random forest and neural network classifier

4 Betting Strategy and Performance

We now construct betting strategies using the learned model and evaluate its performance on the test data (matches that we have not seen during the model training).

For a match with the feature X and outcome Y, if we know the probabilities of $\mathbb{P}(\mathrm{H}|X)$, $\mathbb{P}(\mathrm{D}|X)$ and $\mathbb{P}(\mathrm{A}|X)$, then we can achieve positive expected payoff by betting on outcome O at the agency who offers an odd larger than $1/\mathbb{P}(O|X)$ for any $O \in \{\mathrm{H}, \mathrm{A}, \mathrm{D}\}$. When there is no betting option with positive expected payoff, we do not bet. When there are multiple betting options with positive expected payoff, we would like to choose the maximal one. However, since we do not know $\mathbb{P}(\mathrm{H}|X)$, $\mathbb{P}(\mathrm{D}|X)$ and $\mathbb{P}(\mathrm{A}|X)$ in practice, we use our learned models to estimate them.

Our previous finding that D is very hard to predict motivates us to consider a binary classification of $\{\mathrm{H}, \mathrm{A}\}$ by only using matches whose outcomes are either H or A. In other words, we aim to estimate the conditional probability $\mathbb{P}(\mathrm{A}|\mathrm{A} \cup \mathrm{H}, X)$ and also $\mathbb{P}(\mathrm{H}|\mathrm{A} \cup \mathrm{H}, X)$, rather than directly estimating $\mathbb{P}(\mathrm{A}|X)$ and $\mathbb{P}(\mathrm{H}|X)$. These estimated conditional probabilities readily yield the estimates of $\mathbb{P}(\mathrm{H}|X)$ and $\mathbb{P}(\mathrm{A}|X)$ provided that $\mathbb{P}(\mathrm{A} \cup \mathrm{H}|X)$ is known. We use the empirical estimate in the training data to estimate $\mathbb{P}(\mathrm{A} \cup \mathrm{H}|X)$, i.e. we rely on $\mathbb{P}(\mathrm{A} \cup \mathrm{H}|X) \approx \mathbb{P}(\mathrm{A} \cup \mathrm{H}|X_{train})$. We remark that training a binary classifier of predicting $\{\mathrm{D}, \mathrm{A} \cup \mathrm{H}\}$ does not provide better estimates of $\mathbb{P}(\mathrm{D}|X)$ than the simple empirical estimator. For comparison, we also consider the three-category classification that jointed learned $\mathbb{P}(\mathrm{H}|X)$, $\mathbb{P}(\mathrm{D}|X)$ and $\mathbb{P}(\mathrm{A}|X)$. We use multinomial and binomial logistic regression, repsectively, to learn the probabilities of a two-category and a three-category classification problem. We use the same 70/30 training/test

split.

Armed with the estimated $\mathbb{P}(\mathrm{H}|X)$, $\mathbb{P}(\mathrm{D}|X)$ and $\mathbb{P}(\mathrm{A}|X)$, we consider three different strategies:

1. We only bet on H and A, with $\mathbb{P}(\mathrm{H}|X)$ and $\mathbb{P}(\mathrm{A}|X)$ jointly learned using two-category classification. Then we can avoid the decision of betting on D since that is hard to predict well. Denote this strategy as HA-bet-HA-Learn.

2. We bet on H, D and A, with $\mathbb{P}(\mathrm{H}|X)$ and $\mathbb{P}(\mathrm{A}|X)$ jointly learned using two-category classification and $\mathbb{P}(\mathrm{D}|X) \approx \mathbb{P}(\mathrm{D}|X_{train})$. Denote this strategy as HDA-bet-HA-Learn.

3. We bet on H, D and A, with $\mathbb{P}(\mathrm{H}|X), \mathbb{P}(\mathrm{D}|X)$ and $\mathbb{P}(\mathrm{A}|X)$ jointly learned using three-category classification. Denote this strategy as HDA-bet-HDA-Learn.

For simplicity, we bet one unit on the option with largest positive expected payoff and do not bet if there is no option with positive expected payoff for all matches.

As shown in Table 6, the HDA-betting strategy has higher overall payoff and higher positive match ratio than the HA-betting strategy. That is because the former allows to bet on D using $\mathbb{P}(\mathrm{D}|X)$, even though the used $\mathbb{P}(\mathrm{D}|X)$ is a very rough estimate. Another interesting phenomenon is that if we directly use three-category classification to estimate $\mathbb{P}(\mathrm{H}|X), \mathbb{P}(\mathrm{D}|X)$ and $\mathbb{P}(\mathrm{A}|X)$, the payoff and the positive bet ratio get much worse, even though it is slightly better, in terms of classification accuracy, than the two-category classification with empirical estimate. We also note the good classification accuracy for a category does not necessarily mean good betting payoff. As we see, more payoff can be achieved among matches with outcome A than that among matches with outcome H , while our model has better accuracy for H than A.

Table 6: Betting strategy payoffs evaluation among 4350 test matches.

	HA-Bet-HA-Learn			HDA-Bet-HA-Learn			HDA-Bet-HDA-Learn		
Overall Mean payoff	+1.6%			+5.4%			−0.1%		
Positive Payoff Ratio	28.0%			29.3%			22.3%		
Match Outcome	H	D	A	H	D	A	H	D	A
Positive payoff Counts	740	0	476	725	86	463	273	19	680
Negative payoff Counts	1244	1110	780	1259	1024	793	1711	1091	576
No bet Counts	115	47	29	9	7	5	2	3	3
Mean payoff	+12%	−96%	+71%	+5%	−57%	+61%	−42%	−94%	+148%

Future improvement can happen in designing more sophisticated betting strategies which considers the risk of each betting rather than simply picking the bet with maximal positive payoff. The risk may refer to variance of the payoff, which can be obtained through the variance estimate of the probability estimates.

References

[1] Alan Agresti. *Categorical data analysis*, volume 482. John Wiley & Sons, 2003.

[2] Xin Bing, Florentina Bunea, Yang Ning, and Marten Wegkamp. Adaptive estimation in structured factor models with applications to overlapping clustering. *The Annals of Statistics*, 48(4):2055 – 2081, 2020.

[3] Peter Bühlmann and Sara van de Geer. *Statistics for High-Dimensional Data*. Springer-Verlag Berlin Heidelberg, 2011.

[4] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.