BEAT THE BOOKIE

A PREPRINT

Group Name: `Team C`
Department of Computer Science
University College London
London, WC1E 6BT

December 22, 2021

1 Introduction

It has long been the ambition of man to take the gamble out of gambling, but alas without any success. However, the recent availability of large-scale data and sophisticated machine learning (ML) algorithms may provide a way to at last beat the Bookmakers. No wonder the popularity of attempts to successfully predict sports outcomes has seen a sharp increase in recent years [1].

In this project, we will attempt to predict the outcome of English Premier League (EPL) matches. Research has shown that football is a very difficult sport to predict given that it is team based, has 3 possible outcomes and goal scoring is sparse [1]. Furthermore, past attempts to predict EPL matches such as [2, 3], vary drastically in methodology and results obtained. Many, such as [4], also claimed to achieve extremely high accuracy, over 90%, though these were proven to be unreliable due to their testing process. Indeed, research on predicting football outcomes is extremely nonuniform and yielded no solid foundation on which to approach the problem in a single, decisive manner [1]. This problem demanded careful examination and a suitable approach to not only gain good results, but also reliable results.

Through domain knowledge and analysing previous literature on this topic, we made informed decisions on each stage of this project. Data analysis, as well as our own contextual understanding of the problem task, was combined with sophisticated data transformation techniques to produce a novel, high quality dataset. Multiple classes of cutting-edge ML algorithms and suitable model selection techniques were employed to find the best model for our data. By the end of the project, we were able to match the predictive power of the Bookmakers, achieving 52.7% with a convolutional neural network (CNN).

The paper adopts the following structure: we will first describe our exploration of the data as well as the transformations applied to it (Section 2), followed by an overview of our ML methodology (Section 3) and a more detailed discussion surrounding ML model training and validation (Section 4). Our results are then analysed in Section 5 and our predictions on the test set given in Section 6. We conclude our research and avenues for future research in Section 7.

2 Data Transformation and Exploration

2.1 Data Analysis

Any ML projects requires a granular understanding of the data at hand as well as a contextual understanding.

We first computed 'Summary Statistics', a total of all the existing features as well as the win/loss/draw ratio for each team. Much of the information was unsurprising; those who scored more goals, had more shots on target etc performed better. Interestingly, some teams performed better with fewer goals: Man United had a higher win rate than Arsenal even though Arsenal had scored more goals. Arsenal outperformed Man United on most statistics barring the odd few, most notably the total number of fouls (where Man United had more fouls than Arsenal). This suggested that the more aggressive the team is, the more likely they are to win. It also showed that win rate isn't wholly determined by number of goals scored as one might naively think.

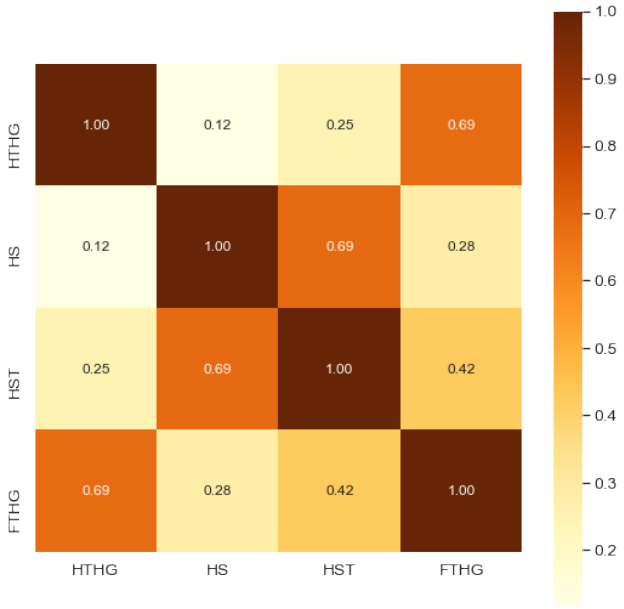

Figure 1: Correlation matrix for FTHGs.

We then analysed the correlation matrix for FTHG compared to other features (Figure 1). Unsurprisingly, the highest correlated features were HTHG, HST and HS (same for FTAG). This informed our decisions on what custom features to engineer (see below).

2.2 Feature Selection and Importance Analysis

Past research has shown that feature selection is an important part of the process and has been shown to boost model performance [1]. We therefore wanted to explore the impact and perform feature selection. Mainly, this was all the features except 'giveaway' features – features who revealed the winner such as FTHG. Before fitting a model, we one hot encoded the Team Names and Referee features. Dates were split into year, month, week and day as required. However, year was removed since we needed to predict future data; there would be no data with a future year (such as 2022 which is the year for data in the test set) hence hindering performance for future predictions. Labels were label encoded and the data was split into train and test sets of size 70% and 30% respectively.

We chose random forest (RF) as the model to explore feature importance and perform feature selection. RF not only produced fruitful results for past projects [1], but it also had useful attributes such as gini impurity [5] which would allow us to clearly analyse the impact of different features. This would help us gain a better understanding of our data and inform our decisions when it came to feature engineering. Furthermore, it meant further data pre-processing, such as scaling, was not required which saved time. Additionally, it was a model that could easily be visualised, making it easier to interpret and assess how features were impacting the outcome. Model performance on all the features was 56.6%.

The first feature we wanted to assess was Referee. Given that referees should be unbiased, this feature should be unnecessary. After removing Referee, model performance increased by 0.25% to 56.89%. Although the change is marginal, it shows that Referee was a noisy feature and hence we removed it before doing feature analysis. This also had the advantage of preventing the curse of dimensionality, given that one hot encoding Referee produced over 40 features. The second feature requiring inspection was dates, a random variable that does not have a visible impact on a game. Interestingly, removing dates decreased the accuracy. We therefore decided to keep it for the time being. This provides an interesting area for future research, namely which non-game related features can impact accuracy.

Having tested non-team related features, we then decided to test RF on only the in-game stats. The idea was to see which in-game statistics would be most important if we had access to any of them. We also wanted to remove any noisy in-game stats to produce a purer feature set and inform our decisions for feature engineering. Given the number of features and that they are all correlated, we employed a more advanced technique for feature selection here to judge the most important features. The accuracy on all in-game states was 55.51%.

Figure 2: Calculated importance of features

Feature selection was done using `recursive feature elimination (RFE)`, a 'wrapper' method which recursively prunes less important features through a ranking process, using the ML model accuracy as a metric. This method was chosen over other options such as `SelectFromModel` due to its superior optimisation iteration process. Other options were also considered such as `Univariate Selection` using `SelectKBest` but were not used due to the extra parameters these methods have which would then need to be tuned [6]. The top 8 features returned by `RFE` were `HST`, `HS`, `AF`, `AS`, `HF`, `AST`, `HC`, `AC`. The reduced feature set produced an accuracy of 54.92%, a small decrease of -0.58% compared to using all in-game stats. The reduced feature subset had an interesting effect on the classifications. Precision and recall increased for `Draw`, was unaffected for `Away Win`, but decreased for `Home Win`. This suggests different features are more useful for predicting certain outcomes, an interesting area for future research.

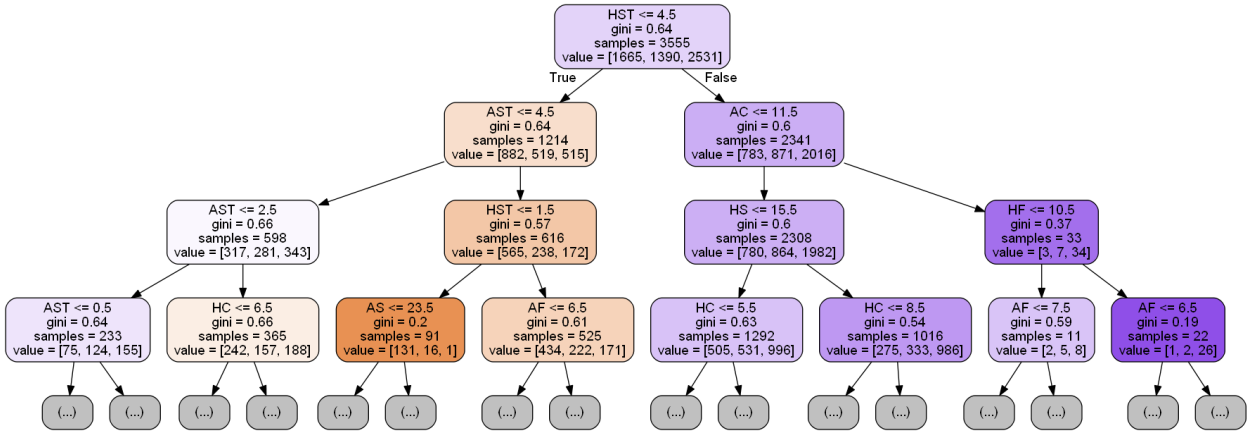

Figure 3: Sample tree displaying how features are used within random forest model.

The different feature importances given by the RF model are shown in Figure 2, showing a steep decline in importance towards the right. A sample tree from the model was used to visualise how the features were being used by the model and is shown in Figure 3.

2.3 Feature Engineering and Data Collection

After gaining a better understanding of the data and using our own intuition of the problem task, we were able to make informed decisions about what custom features to compute. Firstly, we wanted to compute goals / shots on target. This would give us an idea of the skill and scoring strength of the team. For example, a team may have many shots on target, but this is of no use if they are unable to score. Secondly, we wanted to compute shots on target / total shots as this would give the accuracy of a team. Thirdly, given that fouls is an important feature, we decided to compute the ratio of the home teams fouls in the past k matches to away team fouls in the past k matches. This would help assess which

team is more aggressive. Finally, we wanted to compute second half goals scored as currently we only had first half goals scored. We believed the second half was a very important aspect of the match, perhaps even more important than the first half, given that mistakes in the first half can be remedied in the second half but not vice versa. We therefore computed second half goals scored for home and away teams. In addition, we computed the previous matches played, goals conceded, win/loss ratio and cumulative goals scored difference.

Beyond the existing features and those we engineered ourselves, we also wanted to obtain any other relevant data from further sources. Past research has found players attributes, team ratings, attack strength, manager performance streak, managers win etc to be effective features [7, 8]. We were able to find several such features including possession, shooting, team ratings and more from various sources [9, 10]. However, the data was incomplete, going back to only 2017, resulting in the dataset having many missing values. Techniques do exist to impute missing data; however, given that predicting EPL matches is temperamental at best and previous research suggests features should be limited to only the most effective [1], we decided not to use this data. Furthermore, little research has been done on the best way to handle missing data specifically for sports data. However, we were able to find one extra feature without missing data – distance travelled by the Away team [11]. The 'Home Advantage' is a well-known phenomenon in football [1]. Several reasons exist as to why this might be, one of them being fatigue caused by travelling. It therefore stands to reason that the further the distance, the further the fatigue. Hence we believed this could be a useful feature.

Once we had obtained all the required data, we computed prior statistics. In summary, this was the cumulative score for every feature given the past k matches going back to the start of our dataset. This was needed given that for future matches, we would not have access to in-game stats. However, as our dataset would now give historical team strength, it gave little indication as to how strong the team has recently been playing. To provide some balance, we also computed the results and the average shots on goal in the past 3 matches. Finally, the first season of data was removed. This was because to compute the priors, initial matches had stats of 0 which provided no value to the dataset and only added noise. Instead of just removing these initial few rows, we removed the entire first season so that our data began with stats computed using an entire season which we believed would be a more stable dataset for our models to work with.

We now had an extensive, clean dataset with 38 features known to increase performance from past research. We believed this to be a powerful, high-quality dataset which would produce strong results.

2.4 Final Data Pre-Processing

Given that we would be fitting our data using models other than RF, certain pre-processing steps were required which we didn't do before (as they weren't required with RF).

Firstly, dates could not be left as the standard numerical values as the models would assume that a higher date (bigger number) is better where this is not true. Instead of one-hot encoding them, research revealed a better option was to model them as sinusoids [12]. Not only does this prevent an exponential increase in dimensionality, it allows our models to pick up on cyclic trends that may be present such as team performance dropping near the end of every season due to fatigue from the previous matches. This was prudent given that we saw earlier dates positively affect model performance.

Data was split into train, test and validation set using a ratio of 70%, 15%, 15% respectively. Shuffling was disabled to ensure the order of data was preserved which is very important for time series data [1]. However, this does increase the risk of overfitting as there is less randomness in the data. Stratification was not used given that the proportion of each outcome was roughly the same in all segments (45%, 30%, 25% for Home, Away and Draw respectively).

A separate validation set was used as opposed to k-fold cross validation for the same reason of preserving data ordering and therefore maintaining reliability in validation [1, 7].

To prevent overfitting on the validation set, we gathered further data going back to 2000 from [13], increasing the size of the validation set. Combined with using a suitably small function class during model selection, this should mitigate overfitting.

After scaling the data, dimensionality reduction was employed. Although we only had 38 features strictly speaking, our dataset had 120 features due to one hot encoding team names. Past work has revealed the best results are obtained with a small number of features [1]. Dimensionality reduction was used for this rather than feature selection as it allows us to preserve the majority of the information despite the reduction process. A variety of techniques were researched and considered including principal component analysis (PCA), Kernel PCA, linear discriminant analysis (LDA), and t-distributed stochastic neighbor embedding (TSNE) [14]. We found PCA was not effective given that the data is not linearly separable. LDA was also not effective, since it can only find 2 components for our data (given there are only 3 class labels) and is most suited to data that is normally distributed. TSNE was not suitable for our purposes since it is most effective on data with less than roughly 50 features. Kernel PCA, however, with the radial basis function (RBF)

kernel and a default sigma of 1/120 (1/number of features), proved effective, since it transformed the data into a higher dimensional space (making it linearly separable) before applying standard PCA. We chose to reduce the data to 15 features since previous research has shown this to be an optimal choice [1]. Further options such as uniform manifold approximation and projection (UMAP), Isomap and multidimensional scaling (MDS), [15] as well as combinations of techniques such as PCA followed by TSNE, were also explored but did not produce optimal results.

Finally, some models required categorically encoded labels.

3 Methodology Overview

We aimed to test the latest ML models most suited to the task at hand to ensure optimal accuracy was achieved. Many past papers use simple ML models such as logistic regression (LR), support vector machine (SVM) and Naïve Bayes (NB) [2, 3] with good results. Newer research, however, has moved in a different direction. Some papers [16, 17, 18, 19] as well as many top Kaggle submissions [20], make use of boosting algorithms which continuously train weak learners sequentially to correct the mistakes of their predecessor until a learner with sufficient accuracy is built [21]. These have proven to be more effective.

Most papers, however, are now moving towards neural network (NN)s [1]. NNs have seen many advancements in recent years. Architectures have been developed for time series data, such as recurrent neural network (RNN)s. These have an "internal memory" that can remember previous inputs which is ideal for our task [22]. RNNs have also evolved into long short-term memory (LSTM)s and gated recurrent unit (GRU)s which help overcome the vanishing gradient issue RNNs have [22], with several papers utilising LSTMs successfully [23, 24]. Another popular NN which is not traditionally used for time series data are CNNs. However, they have been proven to be rather effective at time series data, due to their innate pattern detection abilities and robustness to noise [25]. Using CNNs for time series data is a growing field and active area of research.

Finally, we have recently seen the rise of time series models such as `Prophet`. These models are good for predicting seasonal trends, and hence a natural fit for our task. Research surrounding them for sports predictions is limited due to their recent invention, but some recent work with them have found relative success [26].

Therefore, we decided to test several classes of ML models which we believed would provide promising results. These consist of Boosting (`XGBoost`, `AdaBoost`, `GradientBoost`, `LightGBM`), NN (Vanilla NN, Deep NN, RNN, LSTM, GRU, CNN – created with `Keras`) and Time Series (`Prophet`, `Arima`) models. These would be compared to "Base Models" (LR, SVM, NB) which would act as the benchmark. In total, we evaluated 15 ML models spanning 4 classes (including Base Models as a single class). Our final approach used our CNN model since it had the highest accuracy.

Our steps for training and evaluation were the following. Firstly, model selection was done using `Optuna` with a suitable function class on the validation set. Optimal parameters were then used for the CNN which fitted on our training set. The model was evaluated using classification accuracy on the test set.

4 Model Training and Validation

4.1 Model Selection

Traditionally, `gridsearch` has been a popular and well-known option for hyperparameter tuning. However, research into this area of ML exposed `gridsearch` to be severely outdated, ineffective and inefficient compared to its newer counterparts [27]. These packages, such as `hyperopt`, provide significant improvements including providing several ways to search the hyperparameter space such as modified tree parzen estimators [28]. We decided to use `Optuna` owing to its sophisticated hyperparameter search toolbox (which included `hyperopt`) as well as universal compatibility which was important for us given the number of ML models we tested [29].

The hyperparameter search space was evaluated on our validation set, which contained roughly 3 seasons worth of data. The function class for each model was restricted to prevent overfitting. For the CNN, we tested different numbers of nodes (ranging from 32 to 256), sigmoid or softmax activation functions and 4 different optimizers (Adam, Rmsprop, Adagrad, SGD) as these were some of the most important hyperparameters [30]. In particular, a good optimizer is very important to ensure the models is trained optimally. Although Adam is overall a very good optimizer given its use of momentum and adaptive learning rate, it is not always the best, hence the need to test several optimizers. The `Optuna` '`study`' was set to maximise the classification accuracy as this was what we would evaluate our model by on the test set.

4.2 Model Training

The CNN was trained on our training set consisting of roughly 13 seasons worth of data which past research indicated would be more than sufficient [1]. The model was trained by using the built in `fit` method for 20 epochs with the Adam optimizer and the default learning rate of 0.001. Categorical cross entropy was used as the loss function.

4.3 Model Evaluation

The final model was evaluated using classification accuracy on the test set. This was used as it was an easily interpretable measure that provided a sufficient way to compare models (given that the dataset is not severely imbalanced). Furthermore, this is widely used in past research [1], hence allowing us to easily compare our results in the next section. Other evaluation metrics were considered, for instance logarithmic loss which penalises false classifications and F1 score which combines precision and recall. However, classification accuracy was used for the reasons aforementioned [31].

5 Results

5.1 Optimal Model Analysis

Our best model was our CNN, consisting of a single `conv1d` hidden layer with 32 nodes with sigmoid activation and optimized using Adam. This model was able to achieve a high accuracy of 52.7%, very close to the gold standard of 53%. It's training and validation accuracies were very close to the test accuracy, indicating the model was trained well and is able to generalise.

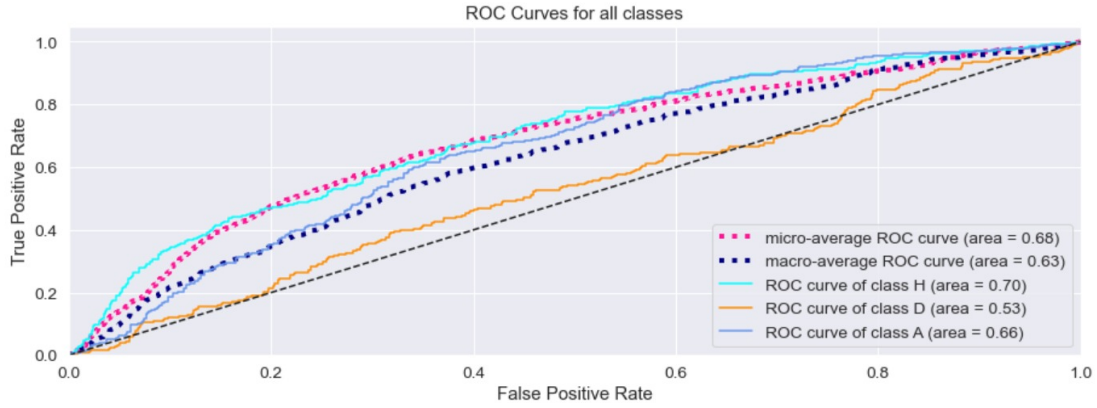

Figure 4: ROC curve comparison across classes.

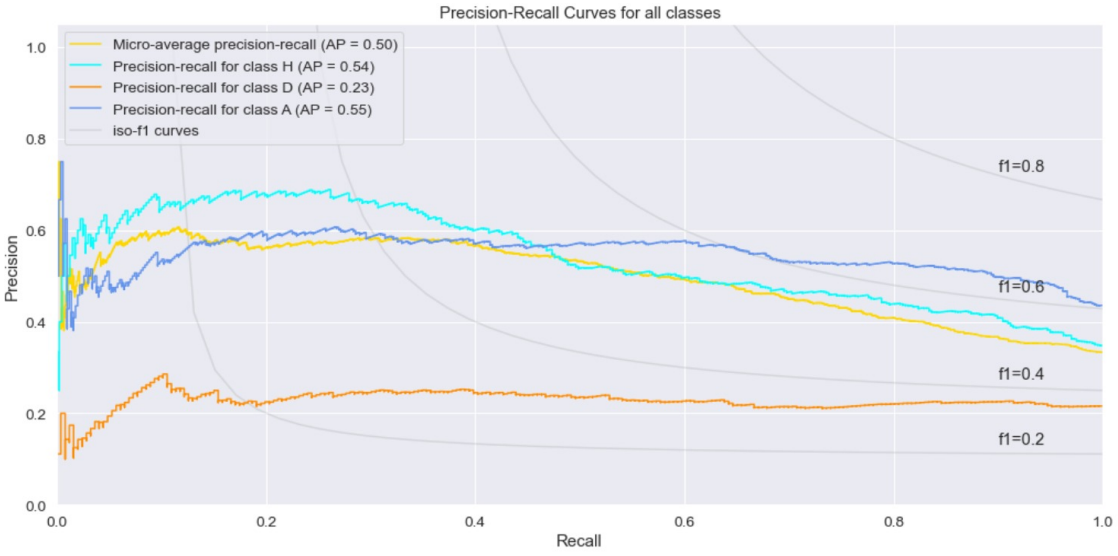

Figure 5: Precision-recall curve comparison across classes.

5.2 Model Comparison

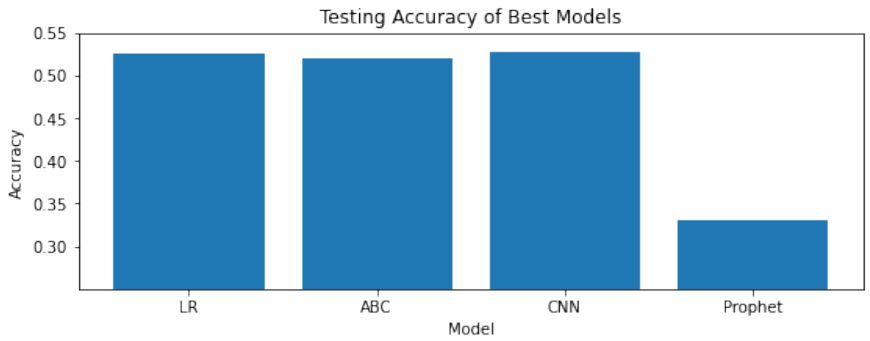

Figure 6: Comparison of test accuracy across models.

Analysing the model's predictions, we can conclude it is good at predicting Home and Away wins but struggles when it comes to Draws. This is shown in the receiver operating characteristic (ROC) curve in Figure 4 which shows the trade-off between sensitivity and specificity. We can see that categories A and H have a relatively good trade off but Draw lies roughly on the diagonal, showing predictions for Draws are effectively random. This is also reflected in the precision-recall curve (Figure 5), where the higher area for Home and Away wins show higher precision and recall compared to Draws. This is an area for future research, analysing why Draws appear to be so much harder to accurately predict.

Comparing the top models from each category (Figure 6), we can make several observations. Firstly, the top Base Model performs very well compared to the more sophisticated algorithms. This suggests that a dataset like ours which is clean and contains useful features can be used with simpler ML models without significantly compromising accuracy. This also has the advantage of reduced model selection and training times as well as an easier process of setting up the model rather than creating one from scratch.

A more interesting comparison of the top models is to compare their Precision, Recall and F1 scores. Figure 7 shows the Precision, Recall and F1-Score of the top model from each class for `Home Win` outcome. Whilst precision and F1-Score are roughly the same for the top 3 models, Boosting algorithms do suffer when it comes to recall. The exact opposite is true for the `Away Win` outcome, however.

Judging models based on accuracy is not sufficient. Analysing the predictions more thoroughly can allow one to make more informed decisions about choosing a model according to one's priorities. For instance, if `Away Wins` provides the highest monetary reward in betting, perhaps a model which can accurately predict `Away Wins` would be preferred over an all-rounder. This is an interesting area for future research.

Analysing the training accuracies does not yield any significant conclusions except that `XGBoost` had a very high training accuracy of roughly 56%. Given that the other models had roughly the same training and test accuracies, `XGBoost` may have been overfitted slightly. This can be solved by increasing the regularization parameter.

5.3 ML Class Comparison

Comparing overall accuracies of each, Boosting models perform the best. This is not surprising given past research has yielded good results with Boosting models. However, the difference is marginal compared to Base Models and NNs. Moreover, as with NNs, Boosting algorithms come with a significant number of hyperparameters which must be tuned carefully.

Base Models performed surprisingly well considering they are much less sophisticated than the other models. As mentioned above, these models have the advantage of being simpler, faster and more reliable to train. Furthermore, such models are generally easier to visualise through XAI techniques compared to the more advanced ML models. Such techniques could provide key information about how to progress in predicting EPL outcomes.

It is interesting to see that the RNN, LSTM and GRU, models developed for time series data, do not seem to provide any advantage over their counterparts. This could suggest that perhaps historical team performance doesn't provide significant value, given that these models which are most suited to such a scenario weren't able to make use of it any more than the other models.

Finally, regarding Time Series models, our results suggest that, while underlying date and time trends may exist in the data (given the accuracy isn't 0), models which focus too much on this may suffer. We believe a better use of these models would be to use them in combination with another ML model; the main ML model could focus on extracting all possible information from team and game statistics and the Time Series model could then factor in seasonal trends.

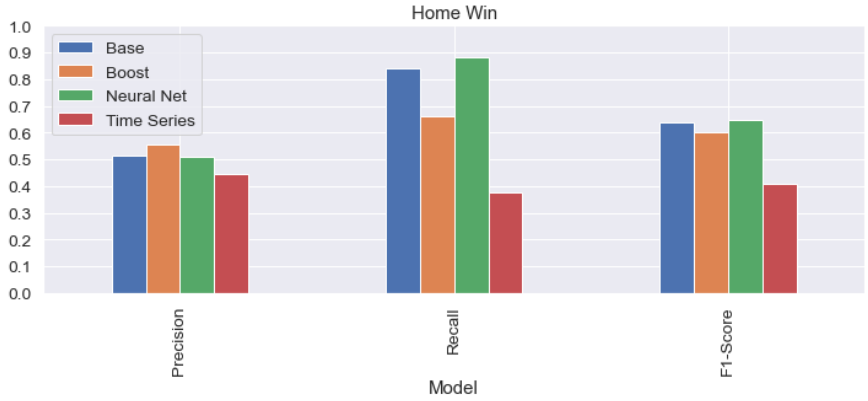

Figure 7: Average precision, recall, and F1 scores across best models in each class.

5.4 Comparison to Past Research

Our results seem to be on par with past research, not being significantly higher nor lower. Examples include [23] who achieved 51% using LSTM and [32] who achieved 58.9%. It was also interesting to note that we achieved roughly the same accuracy as [33], even though our feature set was significantly smaller. This supports research that more features don't necessarily improve accuracy [1].

Some previous studies obtained extremely high accuracy, such as [4] which achieved over 90%. These results, however, are unreliable. This paper uses the same data for training and testing, hence such high results are misleading.

Some papers do achieve higher results than ours which seem valid, such as [18], however they predict other football leagues such as Champion's League. Research has shown that predicting different football leagues can yield vastly different results hence comparisons to such papers is unsuitable [1]. Therefore, we can conclude that our approach, whilst being different from past research, yields suitable results.

6 Final Predictions on Test Set

Date	HomeTeam	AwayTeam	FTR
15-Jan-22	Aston Villa	Man United	A
15-Jan-22	West Ham	Leeds	H
15-Jan-22	Norwich	Everton	A
15-Jan-22	Brighton	Crystal Palace	A
15-Jan-22	Wolves	Southampton	A
15-Jan-22	Liverpool	Brentford	A
15-Jan-22	Tottenham	Arsenal	H
15-Jan-22	Man City	Chelsea	H
15-Jan-22	Newcastle	Watford	H
15-Jan-22	Burnley	Leicester	A

Figure 8: Final predictions on test data.

We applied the same feature engineering and data pre-processing on the Test set as we did on the training set, after which we used our trained CNN model to make our predictions. Our predictions are shown in Figure 8.

7 Conclusion and Future Research

In this project, we aimed to produce a high-quality dataset and apply appropriate ML techniques to predict outcomes of EPL matches, matching the predictive accuracy of the Bookmakers. Through domain understanding, data analysis, feature selection, dimensionality reduction and feature engineering we successfully produced a high-quality dataset going back to 2000. After extensive research of past projects, we were able to shortlist a range of ML models of varying sophistication from which we were able to select the top performing model, a CNN which was tuned through `Optuna` on our validation set, based on its strong classification accuracy on the test set, matching the accuracy of the Bookmakers.

Our project has revealed several avenues for future research. Firstly, external features such as weather should be explored for their possible impact on predictions, given that dates affected our accuracy. Secondly, ensemble models provide a way to overcome limitations of individual models to create an all-rounder. This has been fruitful for many top Kaggle submissions such as [34]. Thirdly, predicting sports outcomes itself is unlikely to be the main aim for individuals or organisations. Custom loss functions can be used to suit the needs of the individual. For example, one project used a custom loss function to maximise the monetary reward from predictions rather than maximise the accuracy [35]. Other avenues include using NLP to analyse situation specific sentiment which has been shown to boost model accuracy by over 6%. [36].

References

[1] Tomislav Horvat and Josip Job. The use of machine learning in sport outcome prediction: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(5):e1380, 2020.

[2] Julian Knoll and Johannes Stübinger. Machine-learning-based statistical arbitrage football betting. *KI-Künstliche Intelligenz*, 34(1):69–80, 2020.

[3] Lalit Kumar Teli, Nilay Zaveri, Pramila Shinde, et al. Prediction of football match score and decision making process. *International Journal on Recent and Innovation Trends in Computing and Communication*, 6(2):162–165, 2018.

[4] Farzin Owramipur, Parinaz Eskandarian, and Faezeh Sadat Mozneb. Football result prediction with Bayesian network in Spanish League-Barcelona team. *International Journal of Computer Theory and Engineering*, 5(5):812, 2013.

[5] Akash Dubey. Feature Selection Using Random forest. `https://towardsdatascience.com/feature-selection-using-random-forest-26d7b747597f`, 2018. Accessed: 2021-12-20.

[6] Pier Paolo Ippolito. Feature Selection Techniques. `https://towardsdatascience.com/feature-selection-techniques-1bfab5fe0784`, 2019. Accessed: 2021-12-20.

[7] Gabriel Fialho, Aline Manhães, and João Paulo Teixeira. Predicting sports results with artificial intelligence–a proposal framework for soccer games. *Procedia Computer Science*, 164:131–136, 2019.

[8] Rory P Bunker and Fadi Thabtah. A machine learning framework for sport result prediction. *Applied computing and informatics*, 15(1):27–33, 2019.

[9] 2021-2022 Premier League Stats. `https://fbref.com/en/comps/9/Premier-League-Stats`, 2021. Accessed: 2021-12-20.

[10] WhoScored. `https://www.whoscored.com/`, 2021. Accessed: 2021-12-20.

[11] Joe Kampschmidt. FootballData. `https://github.com/jokecamp/FootballData/commit/d1903e4c55a222b0507fe5b61402109134ffca88`, 2013.

[12] Satyam Kumar. Stop One-Hot Encoding your Time-based Features. `https://towardsdatascience.com/stop-one-hot-encoding-your-time-based-features-24c699face2f`, 2021. Accessed: 2021-12-20.

[13] Football Data. `http://www.football-data.co.uk/`, 2021. Accessed: 2021-12-20.

[14] Rukshan Pramoditha. 11 Dimensionality reduction techniques you should know in 2021. `https://towardsdatascience.com/11-dimensionality-reduction-techniques-you-should-know-in-2021-dcb9500d388b`, 2021. Accessed: 2021-12-20.

[15] Sivakar Sivarajah. Dimensionality Reduction for Data Visualization: PCA vs TSNE vs UMAP vs LDA. `https://towardsdatascience.com/dimensionality-reduction-for-data-visualization-pca-vs-tsne-vs-umap-be4aa7b1cb29`, 2020. Accessed: 2021-12-20.

[16] Niek Tax and Yme Joustra. Predicting the Dutch football competition using public data: A machine learning approach. *Transactions on knowledge and data engineering*, 10(10):1–13, 2015.

[17] D Buursma. Predicting sports events from past results Towards effective betting on football matches. In *Conference Paper, presented at 14th Twente Student Conference on IT, Twente, Holland*, volume 21, 2011.

[18] Josip Hucaljuk and Alen Rakipović. Predicting football scores using machine learning techniques. In *2011 Proceedings of the 34th International Convention MIPRO*, pages 1623–1627. IEEE, 2011.

[19] Muntaqim Ahmed Raju, Md Solaiman Mia, Md Abu Sayed, and Md Riaz Uddin. Predicting the Outcome of English Premier League Matches using Machine Learning. In *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pages 1–6. IEEE, 2020.

[20] Saif Uddin. Football Match Prediction. `https://www.kaggle.com/saife245/football-match-prediction`, 2019. Accessed: 2021-12-20.

[21] Jocelyn D'Souza. A Quick Guide to Boosting in ML. `https://medium.com/greyatom/a-quick-guide-to-boosting-in-ml-acf7c1585cb5`, 2018. Accessed: 2021-12-20.

[22] Niklas Donges. A Guide to RNN: Understanding Recurrent Neural Networks and LSTM Networks. `https://builtin.com/data-science/recurrent-neural-networks-and-lstm`, 2021. Accessed: 2021-12-20.

[23] Sebastien Goddijn, Evgeny Moshkovich, and Rohan Challa. A Sure Bet: Predicting Outcomes of Football Matches, 2018.

[24] Sarika Jain, Ekansh Tiwari, and Prasanjit Sardar. Soccer Result Prediction Using Deep Learning and Neural Networks. In *Intelligent Data Communication Technologies and Internet of Things*, pages 697–707. Springer, 2021.

[25] Jason Brownlee. How Do Convolutional Layers Work in Deep Learning Neural Networks? `https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/`, 2019. Accessed: 2021-12-20.

[26] Vishnu Nandakumar. Forecasting my beloved Arsenal performance for EPL 2021/22 season using Prophet. `https://medium.com/analytics-vidhya/forecasting-my-beloved-arsenal-performance-for-epl-2021-22-season-using-prophet-ea6167825b11`, 2021. Accessed: 2021-12-20.

[27] Tong Yu and Hong Zhu. Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*, 2020. URL `https://arxiv.org/pdf/2003.05689.pdf`.

[28] James Bergstra, Dan Yamins, David D Cox, et al. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in science conference*, volume 13, page 20. Citeseer, 2013.

[29] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019. URL `https://dl.acm.org/doi/pdf/10.1145/3292500.3330701`.

[30] Suki Lau. A Walkthrough of Convolutional Neural Network — Hyperparameter Tuning. `https://towardsdatascience.com/a-walkthrough-of-convolutional-neural-network-7f474f91d7bd`, 2017. Accessed: 2021-12-20.

[31] Aditya Mishra. Metrics to Evaluate your Machine Learning Algorithm. `https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234`, 2018. Accessed: 2021-12-20.

[32] Alan McCabe and Jarrod Trevathan. Artificial intelligence in sports prediction. In *Fifth International Conference on Information Technology: New Generations (itng 2008)*, pages 1194–1197. IEEE, 2008.

[33] Werner Dubitzky, Philippe Lopes, Jesse Davis, and Daniel Berrar. The open international soccer database for machine learning. *Machine Learning*, 108(1):9–28, 2019.

[34] Gary Lu. Tuning Hyperparameters with Optuna. `https://towardsdatascience.com/tuning-hyperparameters-with-optuna-af342facc549`, 2021. Accessed: 2021-12-20.

[35] Charles Malafosse. Machine Learning for Sports Betting: It's Not a Basic Classification Problem. `https://towardsdatascience.com/machine-learning-for-sports-betting-not-a-basic-classification-problem-b42ae4900782`, 2019. Accessed: 2021-12-20.

[36] Ryan James Beal, Stuart Middleton, Timothy Norman, and Sarvapali Ramchurn. Combining machine learning and human experts to predict match outcomes in football: A baseline model. 2021.