A MACHINE LEARNING APPROACH TO PREDICTING ENGLISH PREMIER LEAGUE MATCH RESULTS

A PREPRINT

Group Name: GROUP G
Department of Computer Science
University College London
London, WC1E 6BT

December 22, 2021

1 Introduction

Europe Premier League football match outcome predictions were computed with 53.54% accuracy by selecting the most accurate out of four models trained and tuned on three different sizes of data sets using cross validation and, for some models, hyperparameter optimization. The four models are Logistic Regression, Naïve Bayes, K-Nearest Neighbours and Support Vector Machines. The most accurate model was the Logistic Regression trained on the data set with variables correlated to game results with a minimum correlation of 0.18 and the hyperparameter C set to 0.2. Moreover, a model that combines the predictions of Logistic Regression and KNN was included. Even though cross-validation accuracy was used as a benchmark, other measures such as confusion matrix, precision, recall and receiver operating characteristic (ROC) curve were used.

2 Data Transformation & Exploration

In order to create a good-performing predictive model, the most relevant variables had to be selected from the collected datasets. To achieve this, raw data relating to team performance or match characteristic were transformed into discrete, continuous, and binary variables. (see full dataset file) A correlation plot with all variables computed in the full dataset was created to identify which variables are most influential on the match outcomes (home win, away win or draw). This resulted into two subsequent datasets, a small and a big dataset. All the models were run using all three datasets to allow for further selection of the best performing combination.

2.1 Data Transformation

To calculate the total road journey travelled by the away team for each match, the spatial coordinates of all stadiums were inputted into FME Workbench. Using the data transformation tools provided by Mark Ireland (Ireland, 2021), the road distance for every potential combination of two stadiums was computed. The team managers and their employment dates have been imported from Wikipedia, and the difference (in days) between their start date and each match date was calculated for both home and away teams. To determine categorical recency, the time intervals were converted into months, and it was assumed that if a manager has been assigned less than six months ago, categorical recency is 1, otherwise it is 0. Goal difference at the end of each match and at half-time has been calculated for each match. These values were then used to calculated cumulated average values for the past seasons. Match streaks is another relevant variable that was calculated for both home and away teams, considering all the matches played against all other teams across all seasons, prior to the present match. Three variables were computed for each team: win, lose, and draw streaks. Columns A to M incorporate singular values for a series of variables that describe the team profile in terms of its players, for each season. The raw dataset sourced from http://www.football-data.co.uk contained variables from all the matches played in the 2008-2021 seasons. This data was then computed in order to create four new categories of variables:

- Home team as home (cumulated averages for all the matches where the home team played home);
- Home total = home team as home + home team as away (cumulated averages for all home team matches);
- Away team as away (cumulated averages for all the matches where the away team played away);
- Away total = away team as away + away team as home (cumulated averages for all away team matches).

The home and away team average age and average market value have been calculated by collecting the players' data for each season from Transfermrkt (Football transfers, rumours, market values and news, 2021) based on the players that played in each match according to Premier League's official website (Premier League Football News, Fixtures, Scores Results, 2021). These four variables, goal differemce and the raw dataset were used to calculate the cumulated average values for all teams in columns N to DI. Because the current season (2021-2022) has not ended yet, the corresponding values were only calculated up until 5th December 2021. Columns N to BM of the dataset contain the cumulated average values for all variables representing the performance of a given team against all other teams accumulated in a season. This variable takes into account both the matches in which the considered team played as home team and away team, respectively, thus considering the four categories mentioned above. In contrast, columns BN to DI of the dataset contain the cumulated average values for all variables representing the performance of the home team against the particular away teams for all previous matches, accumulated across all past seasons. Hence, this offers an overview of all previous interactions between any two teams. The four categories were considered again. Table 1 displays all the data included in the full clean dataset (123 columns), after removing per match values and variables such as date or season:

Table 1: Model Features and Descriptions

Features	Description
FTR	Full-time result of the match
TotalRoadJourney	Total distance travelled by the “away” club measured as distance between home stadium and match stadium
Active Players Home	Average number of players in the season in the home team
Average Age Home	Arithmetic average of age for home team players in the season
Foreigners Home	Number of non-U.K. players in the home team in the season
Average Player Market Value Home	Average market value of a player of the home team in the season
Team Total Market Value Home	Total average value of players of the home team in the season
Active Players Away	Average number of active players in the season in the away team
Average Age Away	Arithmetic average of age for away team players in the season
Foreigners Away	Number of non-U.K. players in the away team in the season
Average Player Market Value Away	Average market value of a player of the away team in the season
Team Total Market Value Away	Total average value of players of the away team in the season
Days Since Home Manager Change	Number of days the current home team manager has been on the team
Days since Away Manager Change	Number of days the current away team manager has been on the team
HT Manager Recency	If home team manager has been assigned less than six months ago, categorical recency is 1, otherwise it is 0
AT Manager Recency	If away team manager has been assigned less than six months ago, categorical recency is 1, otherwise it is 0
HT W/L/D Streak	Home Team Wins/Lose/Draw Streak
AT W/L/D Streak	Away Team Wins/Lose/Draw Streak
home_avg_”variable”	Cumulated average (for each season until that point in time) for “variable” and “var2”, when the home team was (home + away)
home_avg_”variable”_home	Cumulated average (for each season until that point in time) for “variable” and “var2”, when the home team was home
away_avg_”variable”	Cumulated average (for each season until that point in time) for “variables” and “var2” when the away team was (home + away)
away_avg_”variable”_away	Cumulated average (for each season until that point in time) for “variable” and “var2”, when the away team was away
team_home_avg_”variable”	Cumulated average (in total (all season), until that point in time) for “variable” interactions between same teams, when the home team was (home + away)
team_home_avg_”variable”_home	Cumulated average (in total (all season), until that point in time) for “variable” interactions between same teams, when the home team was home
team_away_avg_”variable”	Cumulated average (in total (all season), until that point in time) for “variable” interactions between same teams, when the away team was (home + away)
team_away_avg_”variable”_away	Cumulated average (in total (all season), until that point in time) for “variable” interactions between same teams, when the away team was away

Where “variables” includes the following 18 per match variables (14 of them are from the epl-train and described there): “goal difference at full-time in view of the home team” (HFTgoal_diff); “goal difference at full-time in view of the away team” (AFTgoal_diff); “goal difference at half-time in view of the home team” (HHTgoal_diff); “goal difference at half-time in view of the away team” (AHTgoal_diff), FTHG, FTAG, HTHG, HTAG, HS, AS, HST, AST, HF, AF, HC, AC, HY, AY, HR, AR.

And “var2” includes the following 4 per match variables: “Home Average Age”, “Away Average Age”, “Home Average Value”, “Away Average Value”.

Additionally, a series of assumptions have been made to compute the variables for January matches:

• The distance column will not have bias.

• For the cumulated streaks there will be missing data because some matches have not been played yet. The streaks would represent the streaks until 5th December 2021.

• For the average variables per season, due to the season not being completed, some data might be missing. Therefore, cumulated averages until 5th December 2021 would be considered.

• The bias for the overall historical performance against a specific team across all seasons will be insignificant.

• All average values per season are reset to zero when a new season starts.

• If a team has not played in any of the 2008-2021 seasons, the average team values will also start at zero.

2.2 Data Exploration & Feature Selection

After selecting the initial set of features and transforming them so that they could be used in modelling, dimensionality reduction was employed on the full dataset to create two subsequent datasets of different sizes. This was performed in order to test the models on all three datasets and determine which would perform best. In preparation for feature selection, 'FTM' column was created which would represent the match result as 0, 1, or 2 (home win, draw and away win respectively). Correlations were visualised with a correlation plot of FTM against all other variables. In Figure 1, features which had significant correlation with the match results could be found and should therefore be considered for the subsets.

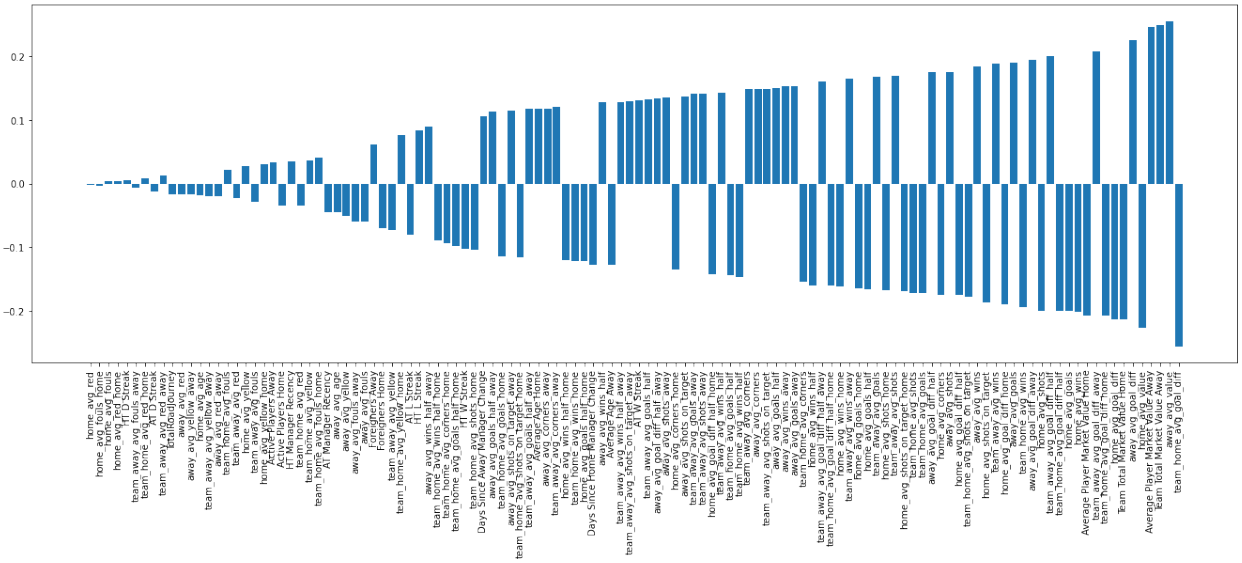

Figure 1: Correlation Plot

For the bigger subset, threshold was set at 0.08, while for the smaller subset, it was set at 0.18 (for each set, any features with correlations below the threshold were excluded). The 3rd set was the original set without any features excluded. The smaller subset was comprised of 24 columns, whereas the larger would contain 86. The smaller subset mainly included team average values and market values, whereas the larger subset had a much broader range of features, notably including variables such as streaks or days since manager change.

Afterwards, 3 new variables were defined (Home, Away and Draw), encoded 0 or 1 and based on the full-time result. Correlations were plotted between these variables and all other predictors and noticed that the feature with the highest correlation for draws had a value of 0.08, whereas for home win or away win the highest value was around 0.2 - meaning that with the current data it would probably be much more difficult to accurately predict draws than wins or losses.

2.3 Missing Values, Data Scaling and Data Splitting

Firstly, there were only 20 rows with missing values (0.3%). Those rows were completely deleted.

Afterwards, the data was split between Y (the match result) and X (all the other variables Min max scaler was used for scaling the X values. This estimator scales and translates each feature individually such that it is in the given range between zero and one.

Finally, the data was split into train and test set (80% and 20% split). The process described above was used for all three datasets.

3 Methodology Overview

Prior to creating the models, a review of the theory and existing literature on the topic has been carried out to establish a suitable framework to train, test and validate the models to then predict football match outputs.

3.1 Additional data sources

To enhance the training dataset for the match-predicting models, several additional data sources were used to collect new variables such as: total road journey (List of Premier League stadiums - Wikipedia, 2021 and The Geography of Football Stadiums (2018): An Example of Data Wrangling and Integration with FME | Safe Software, 2021), manager recency (List of Premier League managers - Wikipedia, 2021), team average age (Football transfers, rumours, market values and news, 2021), team average value (Football transfers, rumours, market values and news, 2021) and other single value variables that describe the team profile in terms of its players (EPL 2021 Payroll Wages Tracker, 2021 and (Premier League 21/22, 2021). These variables do not describe the performance of a team during a match directly but can undoubtedly be correlated with it. Additionally, current season data was missing from the original dataset, so this was collected as well, up until 5th December 2021 (Football Betting, 2021).

3.2 Past Research

To gain an understanding of the different models that could be used as well as how they would be trained, several research papers were analysed to see what approaches to similar problems were taken in the past. Joseph (Joseph, Fenton and Neil, 2006) built a Bayesian Network based on expert judgement and compared it to several other algorithms, including KNN and Naïve Bayes, both of which are models ended up being explored. Hucaljuk et al. (Herbinet, 2018) also looked at these two models in comparison to others, including Bayesian Networks, LogitBoost, Random Forest and Artificial Neural Networks. Hamadani (Hamadani, n.d.) made comparisons between SVM and Logistic Regression with different kernels for predicting NFL match results, which influenced the approach taken in modelling for these algorithms. A paper by Tax et al. (Tax, 2015) which compared several models for predicting Dutch football results was briefly looked at as well, since it additionally summarised some useful variables that were considered for feature selection. Last but not least, Rudrapal (Rudrapal, 2020) used Multi-Layer Perceptron and compared it to SVM, Gaussian Naive Bayes and Random Forest to evaluate their approach.

3.3 Models Overview

All the models created for this project perform multiclass (or multinomial) classification, and the instances are categorized into three classes: Home, Away and Draw. For training, the three scaled data sets made with correlation thresholds, were used. These sets were firstly split by selecting rows randomly with 80% allocated to training and 20% to testing. Cross-validation was also used for evaluation as described below. The models trained were a linear Logistic Regression OVR, a multinomial Naïve Bayes, a k-nearest neighbours, and three types of support vector machines (RBF, Linear (OVR), Linear (OVO)). Additionally, after evaluating the models, another model was computed: a model that uses the predictions of both the Logistic Regression and the KNN.

3.4 Cross-validation and Hyperparameter Optimization

The cross-validation method used to train models split the entire data set into five equal folds to find validation set mean accuracies. Over five iterations, four folds were allocated to a small training data set and the rest to a validation set.

By loop testing through lists of hyperparameters, models such as logistic regression or SVMs were tuned. Indeed, observing accuracies when altering for example, regularization strength parameters for logistic regression or gammas for SVM, enabled the algorithms to be set to perform better.

4 Model Training & Validation

Each model described below was trained on the three datasets resulted from the correlation thresholds, and evaluated using the train, test and cross-validation accuracy. The models were also evaluated using confusion matrices, precision/recall and ROC curves for each class, to compare the performance for every class separately (see section 5).

4.1 Linear Logistic Regression, one-versus-rest

A linear Logistic Regression model was trained. The logistic regression model uses the sigmoid function to transform its output and model conditional probability (Hosseini, 2021). Although the original logistic regression is designed for binary classification, it can be extended to classify multiple classes using the one-versus-rest method. The OVR strategy splits a multi-class classification into one binary classification problem per class. (for example, draws vs rest (home wins and away wins) or away wins vs rest).

The hyper parameter used for optimisation, C, corresponds to an inverse regularization strength parameter. Indeed, models tend to overfit when setting a high C as it gives more importance to the training data. On the contrary to avoid overfitting setting a low C regularises the training data. C values ranging from 0.01 to 10 were firstly tested but the model performed better around 0.1 which led to a refined search for values around 0.1. Figures 2 to 4 display the accuracies on train set, test set and cross validation against our selected values of C, for each dataset.

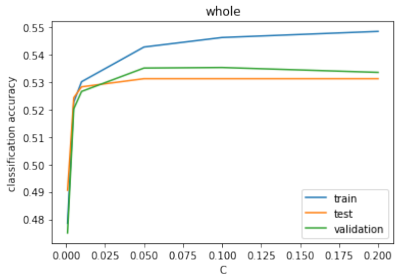

Figure 2: LR Full Data

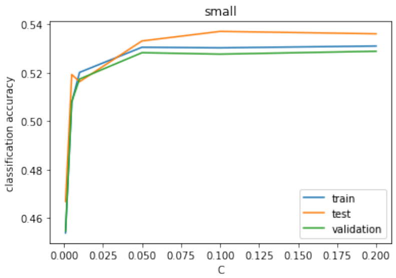

Figure 3: LR Small Data

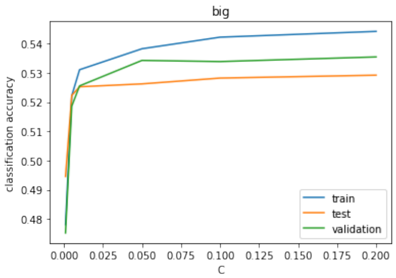

Figure 4: LR Big Data

When plotting accuracies, it was observed that for C values over 0.02, training accuracy kept increasing, while validation accuracy was decreasing, indicating that the model was overfitting. Thus, the hyperparameter selected to train models can be found where the validation set accuracy curve flattens (Hosseini, 2021).

4.2 Naïve Bayes Models

Another model trained was a naïve bayes algorithm which uses Bayes theorem to calculate the probability of an output given an input. Using a gaussian type of naïve bayes model assumes that features follow a normal distribution. However, the output comprises 3 categorical labels and thus, a multinomial type is more adapted (Hosseini, 2021).

4.3 K-nearest Neighbours Models

The classification K-Nearest Neighbours algorithm uses the Euclidian distances between a data point to find its k nearest neighbours and then assigns the class of the majority of its neighbours (Harrison, 2018). For simplicity purposes, hyperparameter n was not optimised and a random number of neighbours was chosen, equal to 7. Consequently, after cross validation, accuracies showed that the KNN fit very well to the training data but not to the validation set indicating a tendency to overfit.

4.4 SVM – RBF, OVR, OVO

Support Vector Machines (SVMs) are classification algorithms that seek to find a hyperplane in an N-dimensional space which can classify the data points, such that the margins of the system can be maximised (Support Vector Machine Algo- GeeksforGeeks, 2021). Although the original SVM is designed for binary classification, the idea of a separating hyperplane can be extended to classify multiple classes. Kernel functions are involved to map the initial dataset into the higher dimensional space and the size of the hyperplane depends on the number of features.

Three different approaches have been used here, Radial Basis Function (RBF) kernel, and two linear kernels with One-vs-One (OVO) and One-vs-All (OVR) classifications. The RBF strategy creates non-linear combinations of the features to uplift the samples onto a higher-dimensional feature space where a linear decision boundary separates the classes (Raschka, 2021). The OVO strategy splits a multi-class classification into one binary classification problem per each pair of classes (for example draws vs. away wins or home wins vs. away wins) The OVR strategy splits a multi-class classification into one binary classification problem per class. (for example, draws vs rest (home wins and away wins) or away wins vs rest) (Hosseini, 2021).

Figures 5 to 13 display the accuracies on train set, test set and cross validation against our selected values of gamma, for each dataset. Before initialising the models, hyperparameter gamma, which defines how far the influence of a single parameter reaches with low values meaning ‘far’ and high values meaning ‘close’, had to be chosen (Kumar, 2020). At first, it was decided to take gamma values of 2^n, where n is equal to the 15 equidistant values in the -10, 10 interval, but the resulting outputs of the RBF models were too flat. Hence, smaller gamma values have been chosen with n equal to the 15 equidistant values in the -20, -4 interval. When it comes to the OVO and OVR, gamma could not be optimized because the outputs remained horizontal lines regardless the interval.

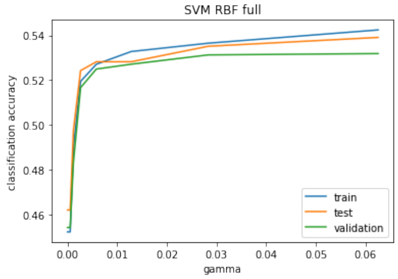

Figure 5: RBF Full Data

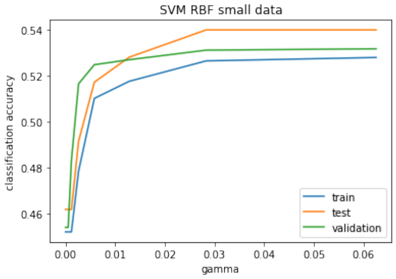

Figure 6: RBF Small Data

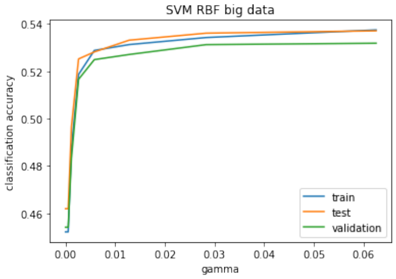

Figure 7: RBF Big Data

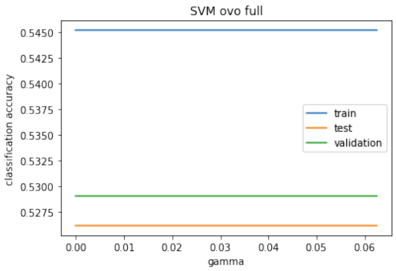

Figure 8: OVO Full Data

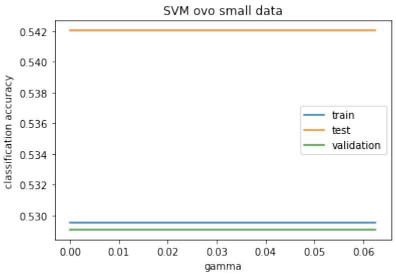

Figure 9: OVO Small Data

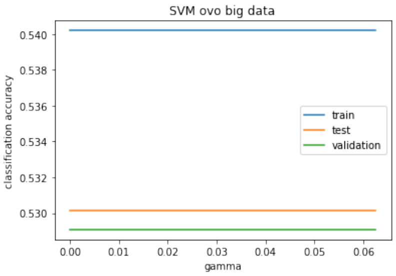

Figure 10: OVO Big Data

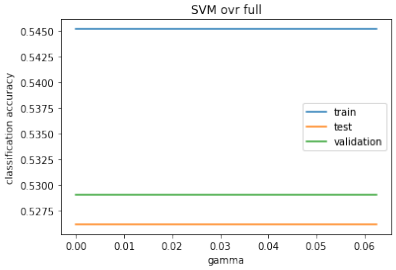

Figure 11: OVR Full Data

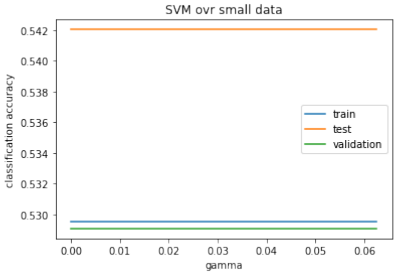

Figure 12: OVR Small Data

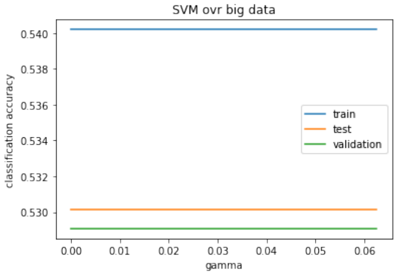

Figure 13: OVR Big Data

4.5 Logistic Regression and KNN

Findings in 5.2. suggest that KNN is the only model that predicted draws. The draws precision was 30% compared to 0% for the other models. Therefore, a model was engineered that uses the predictions of the Logistic Regression in 5.2. (the best model, predicting on the small dataset with hyperparameter C = 0.2), and the KNN in 5.2. (predicting on small dataset). This model computes the predictions of both logistic regression and KNN. The final predictions are the draws predicted by KNN, and the rest of the predictions predicted by Logistic Regression.

5 Results

5.1 Table of accuracies

Table 2: Final EPL Match Predictions

Item No.	Predicting Model	Train Accuracy	Test Accuracy	Validation Accuracy
1	LR full	0.548355	0.531157	0.535216
2	LR small th	0.531041	0.529179	0.528885
3	LR big th	0.544150	0.537092	0.535415
4	NB full	0.519169	0.525223	0.518795
5	NB small th	0.518922	0.523244	0.518399
6	NB big th	0.516201	0.520277	0.517607
7	KNN full th	0.609943	0.450049	0.454687
8	KNN small th	0.606480	0.472799	0.461615
9	KNN big th	0.619589	0.486647	0.481400
10	SVM rbf full	0.542419	0.539070	0.531852
11	SVM rbf small th	0.537472	0.537092	0.531852
12	SVM rbf big th	0.528073	0.540059	0.531852
13	SVM ovo full	0.545140	0.526212	0.529082
14	SVM ovo small th	0.540193	0.530168	0.529082
15	SVM ovo big th	0.529557	0.542038	0.529082
16	SVM ovr full	0.545140	0.526212	0.529082
17	SVM ovr small th	0.540193	0.530168	0.529082
18	SVM ovr big th	0.529557	0.542038	0.529082
19	LR + KNN big th	0.575338	0.524861	0.499122

*th stands for threshold; big th is equivalent to small dataset and small th is equivalent to big dataset.

In total, 19 models have been created and executed in order to select the best-performing predicting model for the EPL matches. After performing cross validation and selecting the optimal hyperparameter for each cell (the hyperparameter that maximises the accuracy for train, test and cross validation), accuracies between models trained on the full, small (0.18 correlation threshold) and big (0.08 correlation threshold) datasets were compared in Table 2.

It can be observed that the accuracy of the models does not follow the same trend across all three columns, and such, the best performing train model is not necessarily the best performing test or cross-validation model. It is worth noticing, nonetheless, that similar mathematical models yielded similar accuracies across all three sets. Moreover, the most desirable approach is to observe the model performance on a validation set, because the bias is lower than that of train and test accuracies. As a result, the best performing model is the Logistic Regression on the Small Dataset (with a threshold of 0.18), which achieved the highest cross-validation accuracy: 0.5354.

It is noticeable that the K-nn had 61% accuracy on the training set, even though it had only 48% accuracy on the cross-validation set. This suggests the model was highly overfitting – this might be due to the fact that 7 was used as the number of neighbours and there were over 5000 observations in the dataset (7 was too small). It is also noticed that the OVO and OVR approaches always have the same accuracy. For the last model (knn + logistic regression) the accuracy did not improve compared to logistic regression, it decreased. This might be due to the fact that a lot of the draws predicted by KNN were false. (as seen in the next section the draws precision is only 30%).

5.2 Confusion matrices

For this section, each model was selected along with the dataset that yielded the best cross-validation accuracy. The optimal hyperparameters shown below were selected using the plots in section 4 (where the curve flattens).

Table 3: Top Performing Models within each Set

Model	Dataset	Hyperparameter
Logistic Regression	small	C= 0.2
SVM RBF	small	Gamma = 2^5
SVM OVO	small	Gamma = any
SVM OVR	small	Gamma = any
KNN	small	-
NB	full	-

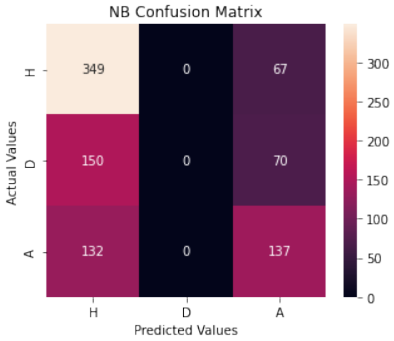

Figure 14: NB Conf. Matrix

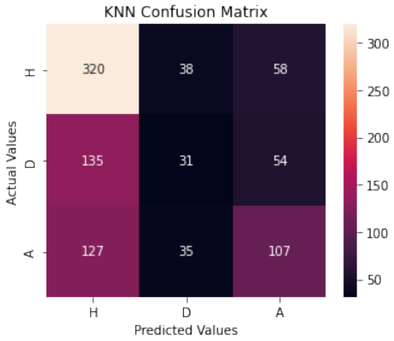

Figure 15: KNN Conf. Matrix

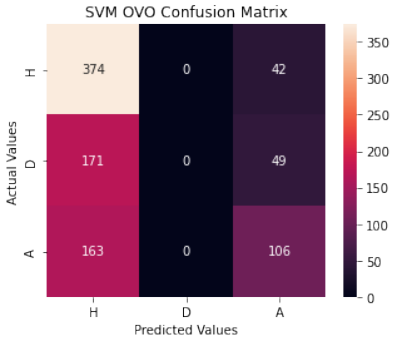

Figure 16: SVM OVO Conf. Matrix

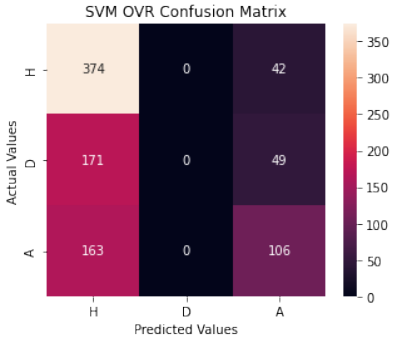

Figure 17: SVM OVR Conf. Matrix

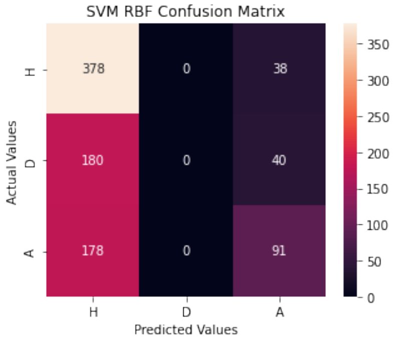

Figure 18: SVM RBF Conf. Matrix

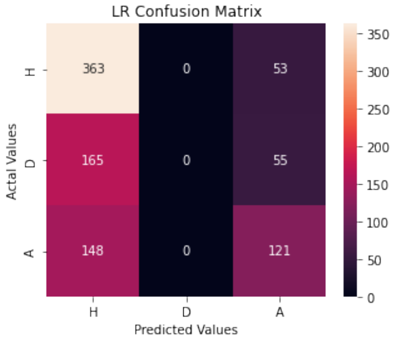

Figure 19: LR Conf. Matrix

It is observed that all the models except KNN predicted 0 draws.

5.3 Precision/Recall

Table 4: Precision/Recall for selected Models

Measure	Logistic Regression	SVM RBF	SVM OVO	SVM OVR	KNN	NB
Precision H	0.53	0.51	0.53	0.53	0.55	0.55
Recall H	0.9	0.91	0.9	0.9	0.77	0.84
Precision A	0.54	0.54	0.54	0.54	0.49	0.5
Recall A	0.39	0.34	0.39	0.39	0.4	0.51
Precision D	0	0	0	0	0.3	0
Recall D	0	0	0	0	0.14	0

Classification report function in Python was used for the above table. The best score for each measure (row) was colored in green. KNN had the highest precision on draws (30%), and the rest of the models 0. The Logistic Regression (the model with highest accuracy score), had the highest precision on away wins and the second highest precision on home wins (the difference was only 0.01).

Table 5: Number of occurrences (in percentage) for each class in the train data (80% of train):

Draws	Away Wins	Home Wins
24%	28%	46%

The table above makes sense because in each confusion matrix, the number of predicted values for draws is either 0, or the smallest (just for knn). Moreover, the number of predicted values for home wins is always the largest. This suggests that the unevenly distributed number of occurrences for each class in the train data might cause some bias.

5.4 ROC curve and Precision/Recall Curves

We used the two best models by cross-validation accuracy scores and used the optimal hyperparameters (where the curves in section 4 flatten), as it can be seen in Table 6:

Table 6: Top 2 Best Models used for ROC & Precision/Recall Curves

Model	Dataset	Hyperparameter
Logistic Regression	small	C= 0.2
SVM RBF	small	Gamma = 2^5

ROC and precision recall curves were plotted for two of the best performing models in order to get a better understanding of the accuracies for different classes in the models over different threshold values.

ROC and precision recall curves were plotted for these models in order to get a better understanding of the accuracies for different classes in the models over different threshold values. Notably, the plots suggested that neither model had any skill in predicting draws. This is represented by a diagonal line in the ROC curves and a flat horizontal line in the P/R curves. Both models performed as expected in terms of predicting home or away wins, and both yielded results that pointed to similar model skill in predicting those, with similar areas under the curve for the ROC plots and similar line shapes for H and A classes in the P/R curves. However, it is possible to tell visually that the area under the curve for class H in the P/R curve is greater than the area for class A. This could be caused by a class imbalance, because, as mentioned earlier, home wins constituted 46% of the training dataset, while away wins represented 28%. ROC is more robust to class imbalance, meaning that this isn't visible on those graphs. Additionally, the micro-averages for the ROC curves are higher than the macro-averages, indicating that the models perform well on average, but do not necessarily perform as well for each individual class (which makes sense since the models had no skill in predicting draws).

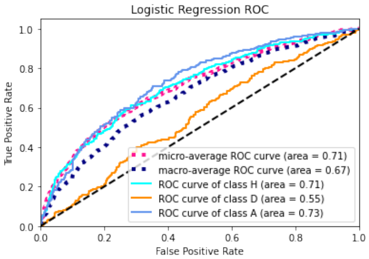

Figure 20: ROC Curve, LR, Small Data

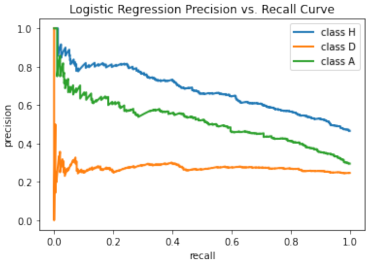

Figure 21: Precision/Recall Curve, LR, Small Data

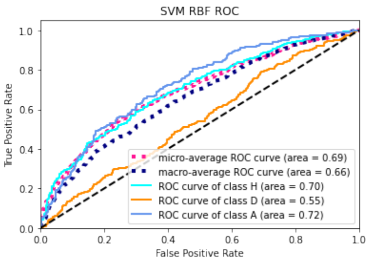

Figure 22: ROC Curve, SVM RBF, Small Data

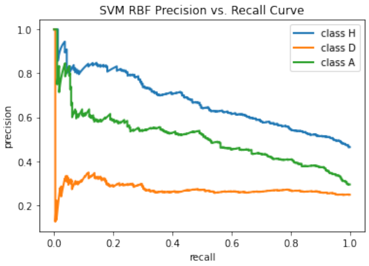

Figure 23: Precision/Recall Curve, SVM RBF, Small Data

6 Final Predictions & Test Set

After choosing the Logistic Regression on the Small Dataset model based on the highest validation accuracy, the best c value hyperparameter had to be obtained for validation. Computation was used again, and the logistic regression was then restrained with the obtained hyperparameter c (0.2). Because the model has already been selected, 100% of the data will now be used for training (compared to the 80% used initially). The EPL January matches columns were computed using the same method explained in section 2.1. Then, the 24 columns in the small dataset were selected and used for predictions. The outputs, originally 0, 1, and 2, were then converted into H (Home win), D (Draw) and A (Away win). The match predictions are listed in Table 7.

Table 7: Final EPL Match Predictions

No.	Date	HomeTeam	AwayTeam	FTR
1	15 Jan 22	Aston Villa	Man United	A
2	15 Jan 22	West Ham	Leeds	H
3	15 Jan 22	Norwich	Everton	A
4	15 Jan 22	Brighton	Crystal Palace	H
5	15 Jan 22	Wolves	Southampton	H
6	15 Jan 22	Liverpool	Brentford	H
7	15 Jan 22	Tottenham	Arsenal	H
8	15 Jan 22	Man City	Chelsea	H
9	15 Jan 22	Newcastle	Watford	H
10	15 Jan 22	Burnley	Leicester	A

7 Conclusion

As bookies tend to get results correct around 53% of the time, the final model is very accurate with a 53.54% accuracy score. However, there are improvements that can be made in terms of bias and performance. Findings in section 5.2. suggest that there might be some biased on the prediction caused by the unevenly distributed number of occurrences for each class in the training data. To address this issue, future research should be pursued to train models on subsets of data with equal number of occurrences for each class. Also, it was mentioned in 1.2. that the correlation between draws and the other variables was the smallest out of all classes, making draws the most difficult to predict. Future research is suggested in engineering other variables that could be more correlated with draws. For example, number of past draws between team x and team y for interaction between same teams. Moreover, performing hyperparameter optimization for KNN could output a more accurate model than the final logistic regression. The reasoning behind this is the accuracy scores on the train set and the fact that knn had the highest precision and recall for draws, as it was the only model that predicted draws. In addition, as accuracies of other models trained, such as SVMs were high and close to the best model's, perhaps training models such as weighted average ensembles which build predictions based on predictions of a selection of models can help reduce bias and can also improve performance.

References

[1] Ireland, M., 2021. The Geography of Football Stadiums (2018): An Example of Data Wrangling and Integration with FME | Safe Software. [online] Safe Software. Available at: <https://www.safe.com/blog/2018/08/fme-and-data-integration-evangelist177/> [Accessed 22 December 2021].

[2] (Football transfers, rumours, market values and news, 2021)

[3] Premierleague.com. 2021. Premier League Football News, Fixtures, Scores & Results. [online] Available at: <https://www.premierleague.com> [Accessed 22 December 2021].

[4] En.wikipedia.org. 2021. List of Premier League stadiums - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/List_of_Premier_League_stadiums?fbclid=IwAR1N-BEka7AcRm8WkIzidx6N73QtyKnSHu46m12T7PLCNmZrHfTDMyYTyN0> [Accessed 22 December 2021].

[5] Safe Software. 2021. The Geography of Football Stadiums (2018): An Example of Data Wrangling and Integration with FME | Safe Software. [online] Available at: <https://www.safe.com/blog/2018/08/fme-and-data-integration-evangelist177/> [Accessed 22 December 2021].

[6] En.wikipedia.org. 2021. List of Premier League managers - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/List_of_Premier_League_managers> [Accessed 22 December 2021].

[7] Spotrac.com. 2021. EPL 2021 Payroll Wages Tracker. [online] Available at: <https://www.spotrac.com/epl/payroll/> [Accessed 22 December 2021].

[8] Transfermarkt.co.uk. 2021. Premier League 21/22. [online] Available at: <https://www.transfermarkt.co.uk/premier-league/startseite/wettbewerb/GB1> [Accessed 22 December 2021].

[9] Football-data.co.uk. 2021. Football Betting | Football Results | Free Bets | Betting Odds. [online] Available at: <http://www.football-data.co.uk> [Accessed 22 December 2021].

[10] Joseph, A., Fenton, N. and Neil, M., 2006. [online] Citeseerx.ist.psu.edu. Available at: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.9659rep=rep1type=pdf> [Accessed 22 December 2021].

[11] Herbinet, C., 2018. [online] Imperial.ac.uk. Available at: <https://www.imperial.ac.uk/media/imperial-college/faculty-of-engineering/computing/public/1718-ug-projects/Corentin-Herbinet-Using-Machine-Learning-techniques-to-predict-the-outcome-of-profressional-football-matches.pdf> [Accessed 22 December 2021].

[12] Hamadani, B., n.d. [online] Cs229.stanford.edu. Available at: <http://cs229.stanford.edu/proj2006/BabakHamadani-PredictingNFLGames.pdf> [Accessed 22 December 2021].

[13] Tax, N., 2015. Tax and Yme Joustra - 2015 - Predicting The Dutch Football Competition Using Pu. [online] studylib.net. Available at: <https://studylib.net/doc/25597619/tax-and-yme-joustra—2015—predicting-the-dutch-footbal...> [Accessed 22 December 2021].

[14] Rudrapal, D., 2020. (PDF) A Deep Learning Approach to Predict Football Match Result. [online] ResearchGate. Available at: <https://www.researchgate.net/publication/335230415_A_Deep_Learning_Approach_to_Predict_Football_Match_Resul [Accessed 22 December 2021].

[15] Hosseini, D., 2021. Logistic Regression Lecture.

[16] Harrison, O., 2018. Machine Learning Basics with the K-Nearest Neighbors Algorithm. [online] Medium. Available at: <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761> [Accessed 22 December 2021].

[17] GeeksforGeeks. 2021. Support Vector Machine Algorithm - GeeksforGeeks. [online] Available at: <https://www.geeksforgeeks.org/support-vector-machine-algorithm/> [Accessed 22 December 2021].

[18] Raschka, S., 2021. How do I select SVM kernels?. [online] Dr. Sebastian Raschka. Available at: <https://sebastianraschka.com/faq/docs/select_svm_kernels.html: :text=The%20RBF%20kernel%20SVM%20decision%20region%20 [Accessed 22 December 2021].

[19] Hosseini, D., 2021. Support Vector Machines Lecture.

[20] Kumar, A., 2020. [online] Available at: <https://vitalflux.com/svm-rbf-kernel-parameters-code-sample/Kernel_Parameter_-_Gamma_Values> [Accessed 22 December 2021].