4741 Project Final Report

Tomas Alvarez (ta352), Benjamin Yeh (by253), Rafael Chaves (rvc29)

1 Introduction

For our project, we plan to use data analysis to answer the following question:

- Can we develop a model that accurately predicts the outcome of soccer games, and could that model be reasonably used in a sports betting portfolio?

2 The Dataset

2.1 Football-Data.co.uk

The primary dataset that we are using to answer our question is the Football Data dataset. This website splits its data into csv files by league, and season (year). For our problem, we decided to make predictions using games from the top five European leagues (England, France, Spain, Italy, Germany) over the last 20 years, giving us about 40,000 examples to work with. Here is a description of the features that this dataset provides:

- **General Match Details**: This includes general information such as the team names of those playing, which team is home/away, the date and time of the game, who the main referee is, spectators in attendance, etc.
- **Match-Specific Statistics**: This includes the half-time and full-time scoreline, number of each team's shots on target, offsides, corners, free kicks, etc. This is information obtained after the match ends, but it is of course not known when betting occurs.
- **Odds**: The standard games odds come in the form of a real number for wins, draws, and losses. As an example, (4, 3.4, 1.95) means that for every dollar a person bets on a home win, draw, or home loss, they get back 4, 3.4, and 1.95 dollars respectively if their result is correct. The dataset also includes odds for Total Goal scored, Asian Handicap odds (a more nuanced betting system), and averages/maximums across all websites.
- **Abbreviations**: The dataset abbreviates all of the column names, which can be confusing for those who are unfamiliar with soccer statistics. For reference, here are what the abbreviations mean for some of the more relevant features:
  - Div: League Division
  - FTHG and HG: Full-time Home Team Goals
  - FTAG and AG: Full-time Away Team Goals
  - FTR and Res: Full-time Result (H=Home Win, D=Draw, A=Away Win)
  - HS/AS: Home/Away team shots
  - HST/AST: Home/Away team shots on target

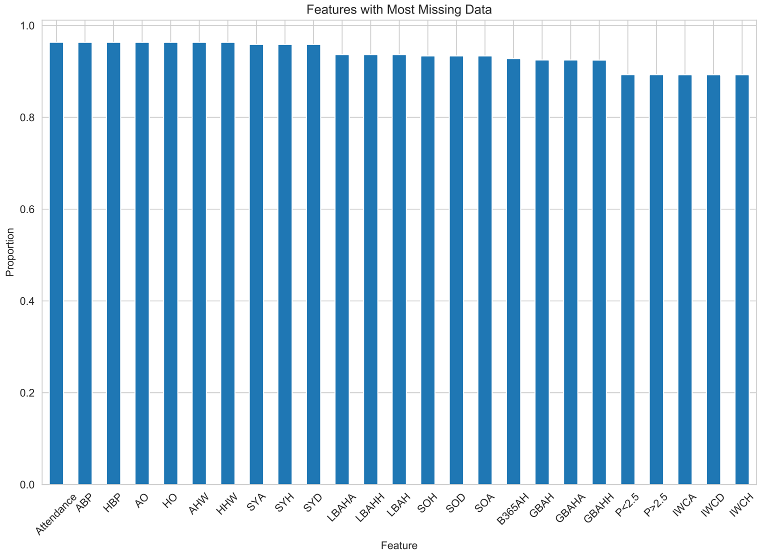
  For a comprehensive list, see https://www.football-data.co.uk/notes.txt

2.2 Club Elo Dataset

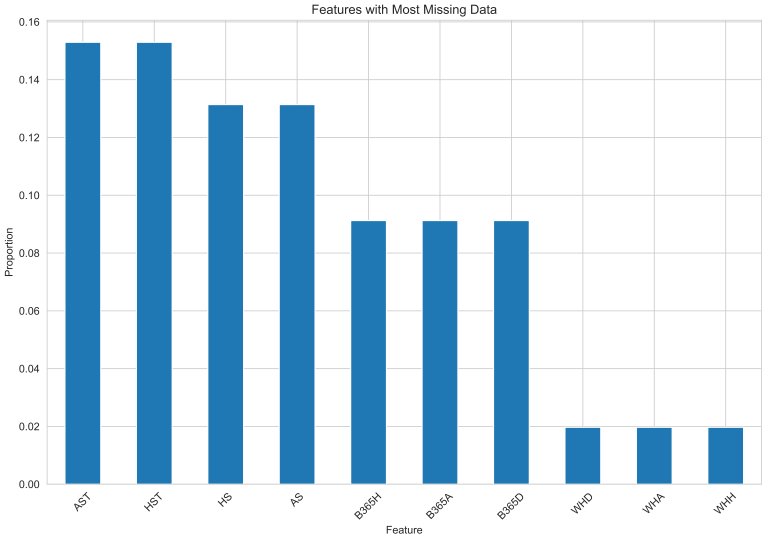
To add another interesting feature, we will augment the primary dataset with data pulled from the Club Elo API. This API provides a full history of any given team's Elo (a measurement of the relative strength of each team) at any point in time. Thus, we can simply query this API and add the Elo of both the home and away team into the dataset. This will provide the model with a very useful feature, as it gives it a sense of the difference in strength between the opposing teams.

2.3 Notable Features of the Data/Missing Values

There are approximately 190 columns in the base dataset with the majority of features pertaining to betting odds from different betting websites. As well, many of the odds are missing for older matches, for reasons such as less data collection in older times, access to historical odds from companies, or companies simply not taking bets for a game. Specifically, of the approximate 7 million entries in the dataframe, there are about 4.5 million NaN values. To further visualize this, we can plot the proportion of missing values for each of the original features:



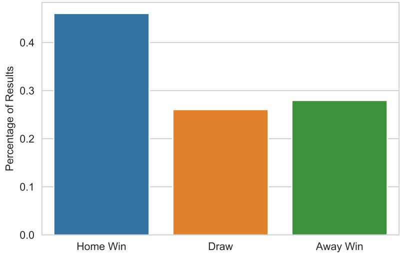
Features with very high rates of missing values will not be useful to us. We subsampled features that we felt were predictive of the outcome of a match, and that also had a relatively small percentage of missing values. These are the `Home Team`, `Away Team`, goals for each team, shots, shots on target, and the odds from the betting companies that had the most (oldest data). With this subsampling, our data is much less sparse:



2.4 Imputation

Although we have some missing values, we can take care of most of them by simply imputing the mean or the mode. We can also use domain-specific knowledge to impute these values; for example, it takes on average 3 shots on target to get a goal, so we can use the full-time result features to predict how many shots on target each time had.

Upon further analysis of the data, we can also inspect how the match outcomes are distributed:

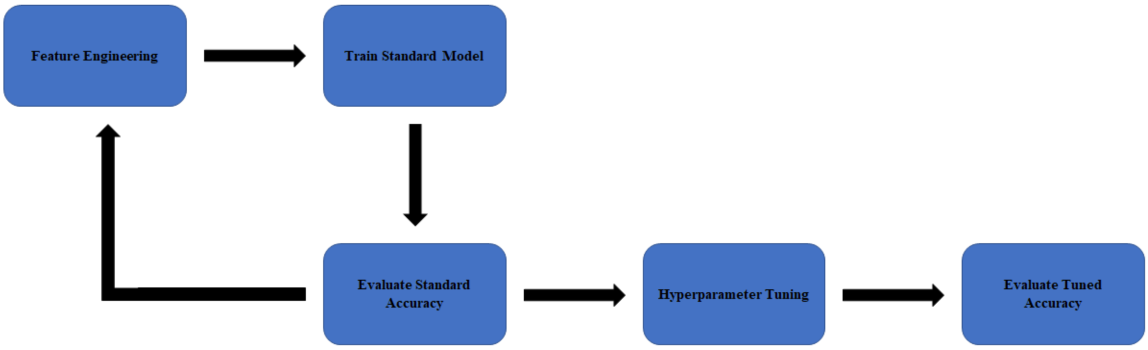


As we can observe, home-field advantage is quite prevalent in the data, with 46% of games resulting in a home team win, versus only 28% for the away team.

3 Modeling Approach

3.1 Framework

The framework for modeling is summarized by the below work flow chart:



Specifically, our design process consisted of iterative feature engineering, training and observing model performance, and then either returning to continue developing/tweaking features or tuning hyperparameters over the current feature set.

3.2 Feature Engineering

3.2.1 Subsampling

Preceding model development, we defined a subset of features to build from: `Division`, `Date`, `HomeTeam`, `AwayTeam`, and `FTR`, the full-time result of the match and target response. We also chose to ignore any available mid-game details, as by our project objective we intend to predict the outcome of a game *before* the match begins, not after it has already started.

3.2.2 Encoding

Because several features in our selection were nominal and non-real values, it was necessary to encode the data into real values so that it could be appropriately passed into a model later. This was done by one-hot encoding the features `Division`, `HomeTeam`, `AwayTeam`.

`Date` however was first decomposed into three features: one indicating the year(e.g., 2000, 2012, etc.), a second indicating the month of the year(1, 3, 10, etc.), and a third for the day of the year(1, 27, 365, etc.). In order to capture the cyclical nature of months and days, a sine and cosine transformation was further applied. Without a sine and cosine transformation, our potential models would be unable to recognize that months '1' and '12' are close together. This would similarly apply to the day of the year feature as well.

The last transformation to encode our dataset to real values was label encoding our response from [`Away, Draw, Home`] ⇒ [0, 1, 2].

3.2.3 Win Streaks

For feature development, the first feature we created was `Home/Away WinStreak`. Within a season, this feature would indicate the winning game streak that the home and away team were on up to the current match. An important note is that this feature only indicated the team's win streak for *home* and *away* matches, not the *consecutive* win streak including both. While simple, the reasoning is that the 'hotter' performing team to match date - be it home or away - is more likely to win the game. In this sense, we are measuring performance between the two teams by how many consecutive home/away games they have won at match time. As observed earlier, there also appears to be a home-field advantage with respect to the outcome of a game. Therefore, capturing whether the home team is also on a home win streak may be a further indicator of the outcome of the match.

3.2.4 Accumulated Wins

We further developed the feature `Home/Away WinsToDate` to capture the accumulated wins for the home and away team within the current season. Similar to `WinStreak`, this feature indicated only the accumulated wins for a team playing as *home* or *away*. It did not capture the total accumulated wins in a season. In a similar manner to `WinStreak`, we hoped to capture in-season performance between two teams by measuring their accumulated wins thus far. While `WinStreaks` captures recent performance, `WinsToDate` is more indicative of a team's 'long-term' home and away performance in a given season.

3.2.5 Last Match Result

For a given match-up between two teams, we created the feature `Last Match Result` to express which of the two teams won the last they played each other. Though certainly not an exhaustive measure for predicting a match, the potential of such a feature is that looking to the result of the past match-up may offer promising insight *especially* if the last facing was very recent.

3.2.6 Elo

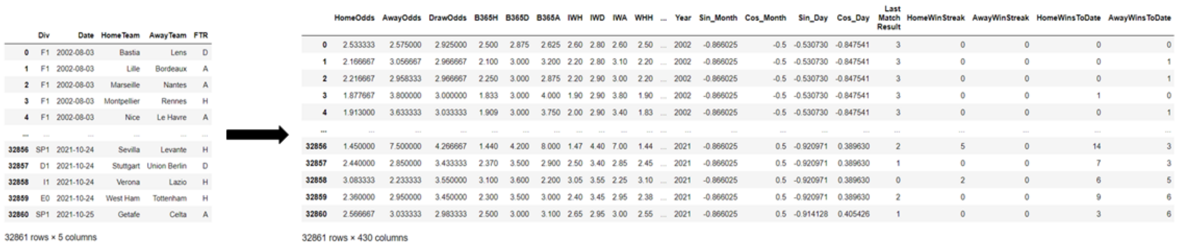
Another feature that we generated was the `Home/Away Elo Score` fetched from the API mentioned above. The API was extremely comprehensive, only possessing missing values for about 5 teams that were no longer in any of the top divisions. To augment this data, we simply queried the API for every unique team in our dataset, stored their complete history into a csv file, and then added the latest Elo of the home and away team prior to the match being played. We feel that this feature will complement the win streak feature nicely, as this should give us a sense of the "favorite" of the match, whereas the win streak captures which teams had played particularly well for the games leading up to the match.

3.2.7 Feature Scaling and Standardization

The final step in feature engineering before moving to build a first model and *after* splitting the dataset was to perform feature scaling and standardization. Following complete feature engineering, our resulting data frame for learning is mostly sparse with the exception of the newly engineered features. It is important to scale and standardize these features appropriately because during training, large magnitude features can dominate over other feature weights and adversely influence gradient update directions as well.

3.3 Engineered Data Frame

Below is the transformation of the dataset before and after feature engineering. The original bare feature space consisted of only 5 columns, and after feature engineering, the resultant data frame had 430.



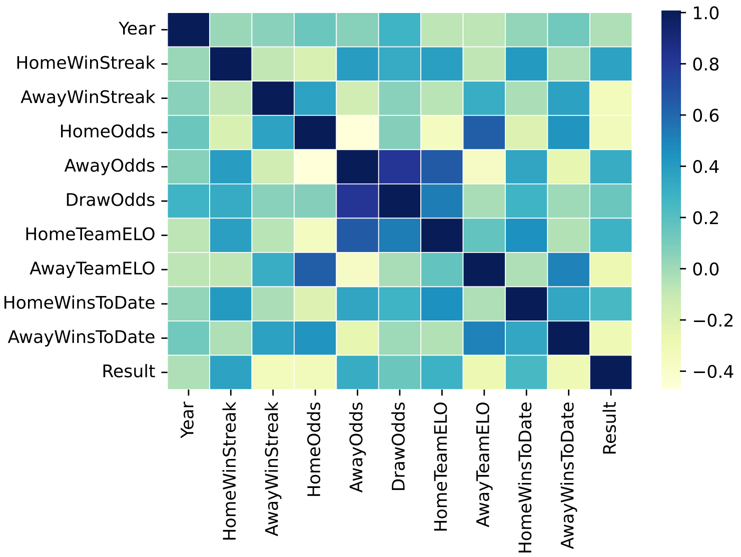
	Div	Date	HomeTeam	AwayTeam	FTR
0	F1	2002-08-03	Bastia	Lens	D
1	F1	2002-08-03	Lille	Bordeaux	A
2	F1	2002-08-03	Marseille	Nantes	A
3	F1	2002-08-03	Montpellier	Rennes	H
4	F1	2002-08-03	Nice	Le Havre	A
...	...	...	...	...	...
32856	SP1	2021-10-24	Sevilla	Levante	H
32857	D1	2021-10-24	Stuttgart	Union Berlin	D
32858	I1	2021-10-24	Verona	Lazio	H
32859	E0	2021-10-24	West Ham	Tottenham	H
32860	SP1	2021-10-25	Getafe	Celta	A

32861 rows × 5 columns

	HomeOdds	AwayOdds	DrawOdds	B365H	B365D	B365A	IWH	IWD	IWA	WHH	...	Year	Sin_Month	Cos_Month	Sin_Day	Cos_Day	Last Match Result	HomeWinStreak	AwayWinStreak	HomeWinsToDate	AwayWinsToDate
0	2.533333	2.575000	2.925000	2.500	2.875	2.625	2.60	2.80	2.60	2.50	...	2002	-0.866025	-0.5	-0.530730	-0.847541	3	0	0	0	0
1	2.166667	3.056667	2.966667	2.100	3.000	3.200	2.20	2.80	3.10	2.20	...	2002	-0.866025	-0.5	-0.530730	-0.847541	3	0	0	0	1
2	2.216667	2.958333	2.966667	2.250	3.000	2.875	2.20	2.90	3.00	2.20	...	2002	-0.866025	-0.5	-0.530730	-0.847541	3	0	0	0	1
3	1.877667	3.800000	3.000000	1.833	3.000	4.000	1.90	2.90	3.80	1.90	...	2002	-0.866025	-0.5	-0.530730	-0.847541	3	0	0	1	0
4	1.913000	3.633333	3.033333	1.909	3.000	3.750	2.00	2.90	3.40	1.83	...	2002	-0.866025	-0.5	-0.530730	-0.847541	3	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
32856	1.450000	7.500000	4.266667	1.440	4.200	8.000	1.47	4.40	7.00	1.44	...	2021	-0.866025	0.5	-0.920971	0.389630	2	5	0	14	3
32857	2.440000	2.850000	3.433333	2.370	3.500	2.900	2.50	3.40	2.85	2.45	...	2021	-0.866025	0.5	-0.920971	0.389630	1	0	0	7	3
32858	3.083333	2.233333	3.550000	3.100	3.600	2.200	3.05	3.55	2.25	3.10	...	2021	-0.866025	0.5	-0.920971	0.389630	0	2	0	6	5
32859	2.360000	2.950000	3.450000	2.300	3.500	3.000	2.40	3.45	2.95	2.38	...	2021	-0.866025	0.5	-0.920971	0.389630	2	0	0	9	6
32860	2.566667	3.033333	2.983333	2.500	3.000	3.100	2.65	2.95	3.00	2.55	...	2021	-0.866025	0.5	-0.914128	0.405426	1	0	0	3	6

32861 rows × 430 columns

Additionally, we can gain a sense of how closely correlated some of these new features are with the match result using a correlation matrix:



We can see that many of these feature transformations that we have performed appear to have a fairly strong correlation to the label. For example, the two features measuring win streaks have respective correlations of about 0.5 and -0.3 with the result. It is also worth noting the correlation between the average match odds and the overall result: they tend to be good predictors of how the match will play out.

3.4 Candidate Models

3.4.1 Home Wins Baseline Model

Noted before, we observed that there is a large advantage for the playing home team. We proposed then a hypothetical baseline model that simply predicted for a given match that the defending home team would always win. This model produced a training accuracy score of 46.49%, and a testing score of 42.85%. These scores served as a benchmark to evaluate against and beat for subsequent models developed. If we were unable to produce a model that predicted better than this baseline, then it would be a clear indication of severe underfitting and/or poor model choice.

3.4.2 Linear Models

For model development, we considered the two linear models: `Logistic Regression` and `Support Vector Machine`, both classic machine learning models for performing classification tasks.

In developing a Logistic Regression model, we considered using both $l1$ and $l2$ regularization. Considering how our feature dimension had exploded in size, we wanted to especially guard against overfitting. As well, with a larger feature space, we were also curious to observe how differences in $l1$ and $l2$ regularization and the resultant weight vectors w would influence the resulting scores.

With building an SVM, in addition to utilizing $l2$ regularization, we also used a non-linear kernel, the radial basis function, or RBF.

3.5 Train-Validation-Test Split

Each model was fit over the 'entire' (considering *all* leagues and teams) dataset, specifically trained over 15 seasons from 2002 to 2016. The 2017 season was reserved for validation scoring for feature engineering, and the remaining 4 seasons 2018 to 2021 were reserved for testing, emulating an approximate 75-5-25 split.

4 Prediction System Results

4.1 Tabled Results

Below are the tabled results of each cycle of developing a new feature and observing its subsequent effect on model performance.

Feature Engineering Results - Training						
Algorithm	No Feats.	WinStreak	WinsToDate	LastMatch Result	Betting Odds	ELO
*l*1 Logistic Regression	0.5217	0.6123	0.5724	0.5217	0.5301	0.5470
*l*2 Logistic Regression	0.5216	0.6127	0.5766	0.5218	0.5319	0.5477
Support Vector Machine	0.5457	**0.6402**	0.46498	0.5263	0.5296	0.5776

Feature Engineering Results - Validation						
Algorithm	No Feats.	WinStreak	WinsToDate	LastMatch Result	Betting Odds	ELO
*l*1 Logistic Regression	0.5067	0.5837	0.5837	0.5059	0.5321	0.5483
*l*2 Logistic Regression	0.5044	0.5836	**0.5891**	0.5046	0.5293	0.5459
Support Vector Machine	0.4954	0.5836	0.43915	0.4949	0.5392	0.5494

As well, below are the results of each final model fit on two feature sets - one complete, and one hand-selected - motivated by the results of the correlation matrix plot and validation scores. Further note that because there was no significant improvement from using all features to the hand-selected set, we did not further perform hyperparameter tuning for those models.

Feature Set Results - Training				
	All Features		Home Streak, Away Odds Draw Odds, Home ELO, Home Wins To Date	
Algorithm	Standard Fit	Tuned Fit	Standard Fit	Tuned Fit
l1 Logistic Regression	0.6325	0.6325	0.5867	—
l2 Logistic Regression	0.6316	0.6316	0.5855	—
Support Vector Machine	**0.6925**	0.6270	0.6109	—

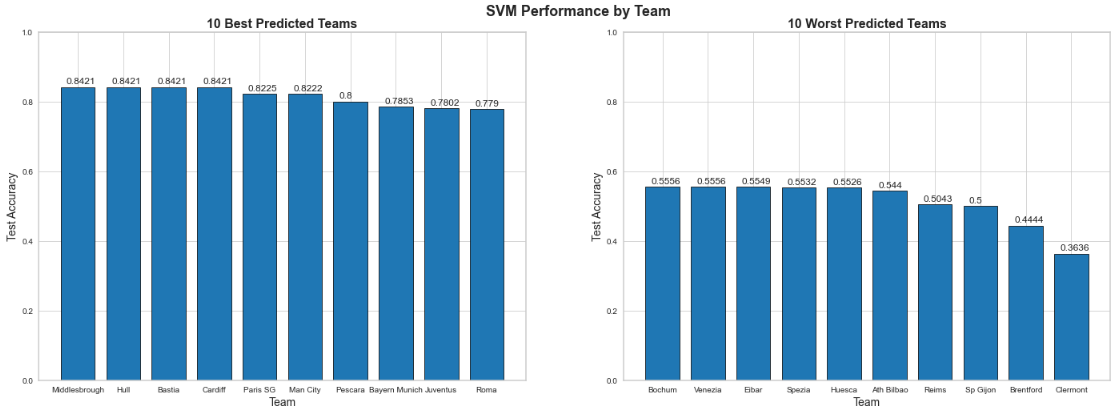
Feature Set Results - Testing				
	All Features		Home Streak, Away Odds Draw Odds, Home ELO, Home Wins To Date	
Algorithm	Standard Fit	Tuned Fit	Standard Fit	Tuned Fit
l1 Logistic Regression	0.6252	0.6278	0.5682	—
l2 Logistic Regression	0.6252	0.6280	0.5634	—
Support Vector Machine	**0.6576**	0.6310	0.5955	—

The most significant feature additions were `WinStreak`, `WinsToDate`, `ELO`, and `Betting Odds`. Further, we noticed that of these features, not all resulted in improved performance across all models. For instance, the addition of `WinsToDate` actually resulted in poorer performance for the SVM, decreasing its validation score from 49.54% to 43.91%, putting it at comparable performance to the `Home Wins` baseline model. We can also observe that the choice of *l*1 or *l*2 penalty did not result in drastic differences in Logistic training, validation, and testing accuracies. What most influenced the scores - across all sets - were the features.

4.2 Best Model Performance

The `Support Vector Machine` trained over all engineered features achieved the best performance, reaching the highest test accuracy score of all models at **65.76**%. This is a sizeable improvement from the baseline `Home Wins` model, which achieved only 43.92% accuracy over the same test set - a staggering 21.84 difference in accuracy. This is a further impressive improvement compared to the baseline SVM with no engineered features which achieved 49.54% on the validation set.

Further, we are also able to interestingly view the breakdown of the SVM's performance by team, visualized below.



Specifically, the SVM predicts best on `Middlesbrough` Football Club matches, reaching an astonishing 84.21% accuracy. Further interesting is that a probe into the distribution of outcomes for matches played by Middlesbrough revealed that they were rather uniform. That is, matches played involving Middlesbrough were not terribly skewed to one outcome, indicating that the SVM was able to fit those particular matches quite well and *not* due to an unfair characteristic of Middlesbrough games(e.g., Middlesbrough matches rarely resulting in a home win, Middlesbrough winning the majority of the matches they play in, etc.).

Conversely, the SVM performed quite poorly for `Clermont` Football Club matches, achieving only 36.36% accuracy. A similar probe into these results produced an alarming but reasonable explanation: Clermont appeared only 11 times - all of which in 2021. I.e., Clermont had not appeared once during training and showed only at test time, to which the SVM would have interpreted as an unknown team. This interestingly demonstrated the SVM's inability to generalize well to new information such as the addition of a team to a division, and would further suggest careful application and consideration of such a model if used in a true setting.

5 Concluding Remarks

5.1 Production Assessment

We had initially set out to develop a model for a betting system. Purely from a machine learning perspective, we are impressed with the achieved performance and specifically the improvement from 43.92% accuracy to 65.76%. However, with 65% accuracy, this would likely lead to more variance than most bettors would be comfortable with. However, given that a firm is tolerant to risk, the model could make money in the extended long run. Therefore in practice, this model would be appropriate for firms/individuals with enough capital to bet in the long run, but not ideal for short-term risk averse firms/individuals. Further, in combination with soccer domain knowledge, we feel confident in deploying our model to enhance a betting portfolio and predict match outcomes.

5.2 Fairness Assessment

We do not believe that there are issues of fairness in this project. Our features are quantitative and measure a team's relative strength based on previous match data. None of these features are protected attributes so our model does not unfairly classify any sensitive information. As well, our model does not inherently yield harmful or threatening consequences, nor does it have a significant impact as the results of games are fully independent of the model predictions. Lastly, the potential risks of trusting the model - by betting and losing money - are already inherent to the nature of gambling. The produced model provides only machine insight to which team is predicted to win, and should not be taken to truth.