An Immovable Object Meets an Unstoppable Force:

Does Defense Produce Success in Football?

Kyle Tan (500564538)
Supervised by Tamer Abdou
4/1/2023

Ryerson University

Table of Contents

Revised Abstract

Abstract:

"Attack wins you games, defense wins you titles", these are the words of one of the most recognized football managers, Sir Alex Ferguson (Smith, 2017). This theory is an often-said rhetoric that spans all sorts of sports (Davis & Suryawanshi, 2023). However, with the scope of this project, this theory will be analyzed through the scope of international football (soccer). It needs to be recognized that the psychology, challenges, and difficulties in each different competition varies quite a bit. League football, for example, takes place over months, with different levels of support, control, time, and accountabilities than cup football. Time has passed since Sir Alex said the famous quote. Since then there has been major developments and changes in the dogma of both club and international football, from the rise of possession seen in Vincent Del Bosques' successful Spain side (MARCA, 2012) to the rise in popularity and success in positional play seen in Pep Guardiola's Manchester City (Breaking the Lines, 2022) all of which leads away from a traditionally "defensive" mindset. Despite the success and popularity of these developments, we saw in the most recent world cup, teams with fantastic defense records like Argentina and Morocco saw success and trophies (FOX Sports, 2023). The question this project aims to answer is: does the popular rhetoric still apply to today's football? Using data provided from the European Soccer Database from Kaggle, this project will aim to use predictive analytics techniques to aim and see if a defensive team means success/higher chances of winning.

Literature Review

Defining Research Questions:

This project aims to answer one simple question: Does defense win titles in the context of international football? With more research done into previously completed projects and works, the question needs be further fine tuned and expanded.

LITERATURE REVIEW AS PER COURSE MODULE:

- What do you already know about the topic?

Personal knowledge of this topic comes from previous attempts at this project and purely rhetoric. From my previous attempts at the project and the extremely simple and poor results, chances are success is created from a balance of both attack and defence not a sole focus on one or the other.

Defensive performance is not a easily definable and measurable item for football. A major challenge would be to create something useable to measure defensive performance. As defensive statistics for soccer is an incredibly complex and understudied area. There has been attempts to develop a sound view to look at defensive by previous experts (Winterburn, 2017) (Brownell, 2013), they all agree on one fact: there is much works to be done, and there's much more than meets the eye.

- What do you have to say critically about what is already known?

Research on defensive performance and machine learning in the field of football is fairly limited, most work is done on the attacking side of the game (Merhej, Beal, Ramchurn , & Matthews, 2021). However, there are researchers that have tackled the topic.

One proposed analysis of looking at defenders and rating defense is through using a GNN (graph convolutional neural network) to sort unstructured data to model certain defensive behaviours (Stöckl, 2021). The researchers trained a GNN model based on real time data using certain predictors they have created. They are able to create some good graphical analysis of defence. However with my limited time and skillset, and the lack of source code on their project, I am unable to create something similar.

Merhej et al.'s research has provided a solid foundation on using deep learning to value defensive actions. By predicting what is going to be stopped they are able to model what they call the DAxT measure (defensive action expected threat).

Critically, what I have to say about the overall topic is that research done is extremely specific and niche. It's also a relatively new subject where there doesn't seem to be a consensus on what is the one true measure for defense.

- Has anyone else ever done anything exactly the same?

There are research into looking at defence in football. But not yet any documents I'm aware of for the usage of seeing if defense truly wins through a data perspective.

- Has anyone else done anything that is related?

Yes, there are a lot of machine learning projects done on the topic of defensive and football. A lot of the work done is specific to other areas of the game like the offensive sides, injuries (Rossi, et al., 2018).

Most of the work are predictive modelling. For example this following paper that looks to use a set of attributes to help predict transfers using Random Forest, Naïve Bayes,and AdaBoost algorithms (Ćwiklinski, Gielczyk, & Choras, 2021). I plan to take a similar approach, whereas their goal is to create a good predictive model. My goal is to look at which variables are most impactful.

The positive side is that there are a lot of machine learning work done, but just not for my specific topic.

- Where does your work fit in with what has gone before?

My work here aims to further explore the field of defense in the context of world football. It will aim further help understand defense and which ones are key to a team's success.

- Why is your research worth doing in the light of what has already been done?

The main goal is to develop a further understanding of the game. No project has really looked at predicting success. Most research in relations to defence is based on creating a good measure/predictor for defence.

Proposed Methodology

Previous attempts and methodology are removed and ignored as database was not large enough.

New proposed methodology using data from
https://www.kaggle.com/datasets/hugomathien/soccer?datasetId=63&sortBy=voteCount

I will load in all match data from the top division in England, France, Germany, Italy, Spain.

The predictor I will use for success will be based on the available betting odds, as low return odds generally mean a favorable match and I will use this as my measure of "success". This method is a very general relation based on predictions made by betting companies.

I hypothesize little link between most of the predictive stats and the chances of success. There are other experts that also have said football is one of the more unpredictive sports (Anderson & Sally, 2013).

A team will be defined as successful it they're predicted to win their matches (defined by low odd returns) and are considered unsuccessful if their victory is unpredictable and less likely (defined by high odd returns).

For the scope of the project, I will be looking at home wins.

For the sake of this project, defense is defined by the traditional sense that a team will look to sit back in their own half, allow the opposition to be on the ball. The main aim not to create and score, but rather not allow the opposition to score. This can be defined as teams with low build up speeds, higher defensive aggression, lower defensive team width, plays less expressively and plays conservatively.

As the original database is a SQLLite file, I will be taking inspiration from: https://www.kaggle.com/code/yonilev/the-most-predictable-league/notebook on how to read in and organize the data. I will not using their entropy analysis but simply how they read in the data

As recommended from project feedback I will be initially using filter-based, wrapper, and embedded/hybrid techniques to do feature selection. Pandas profiling is used for EDA report generation.

STEPS TAKEN:

1) Pre-Processing

- Frist I will generate the average odds as a singular dependent variable. I plan to take an average of all the odds provided by the betting companies.
    i. First issue to come up: Some betting companies have a very high level of missing values

    ii.
- Any variable with more than 1% "missing values" are simply dropped.
    i. 14585 observations prior to dropping columns with missing values
    ii. 14503 observations. after dropping columns with missing values.
    iii. Data loss of less than 1%
    iv. Below is comparison of the missing values % between each betting company

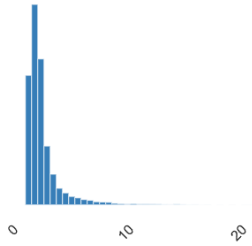
Attribute Name	Missing Values
B365H	0.1%
BWH	0.2%
IWH	0.3%
LBH	0.01%
PSH	50%
WHH	0.01%
SJH	24.1%
VCH	0.2%
GBH	37.7%
BSH	37.7%

v. Once missing data is removed, an “avgOdds” (average odds) variable was created by taking the average of between all odds provided by betting companies for each match.

- Output Variable/dependent variable (avgOdds) is a continuous, numerical piece of data. As such may be hard to build models against. It is converted into a set of categories.

avgOdds
Real number (ℝ)

Distinct	2266	**Minimum**	1.042
Distinct (%)	2.7%	**Maximum**	28.4
Missing	0	**Zeros**	0
Missing (%)	0.0%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	2.5268982	**Memory size**	1.3 MiB



More details

Statistics | Histogram | Common values | Extreme values

Quantile statistics

Minimum	1.042
5-th percentile	1.258
Q1	1.682
median	2.1
Q3	2.684
95-th percentile	5.52
Maximum	28.4
Range	27.358
Interquartile range (IQR)	1.002

Descriptive statistics

Standard deviation	1.6247736
Coefficient of variation (CV)	0.64299131
Kurtosis	23.844992
Mean	2.5268982
Median Absolute Deviation (MAD)	0.478
Skewness	3.8963364
Sum	215094.63
Variance	2.6398891
Monotonicity	Not monotonic

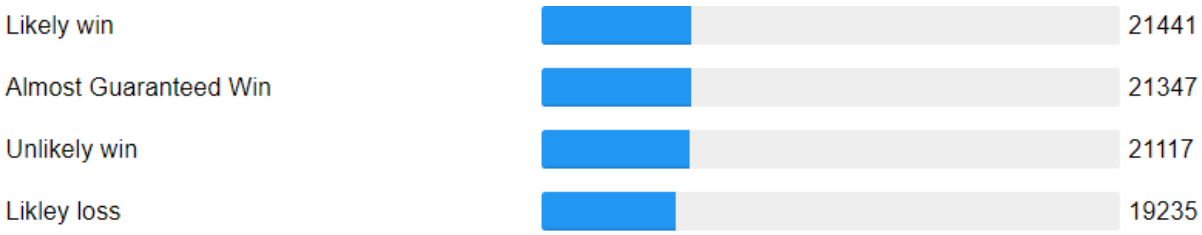
i. Outlier removal and detection was done. Anything outside of 3 z scores are all removed as they can be considered outliers.
   - Ignorable amount of data loss here (dependent observations went from 85122 to 83140, roughly a 3% loss of observations, not a huge deal).
   - Likewise for categorical data, it reduced from 85122 observations to 83140 observations (< 3% data loss)

ii. The continuous numerical data are converted into 4 categories/ From the EDA report, we can see this output data is not evenly distributed. As such the categories should be divided into groups that are representative. (e.g. simply taking the ranging and dividing by 4 would result in a biased grouping where one is weighed much heavier than the rest).

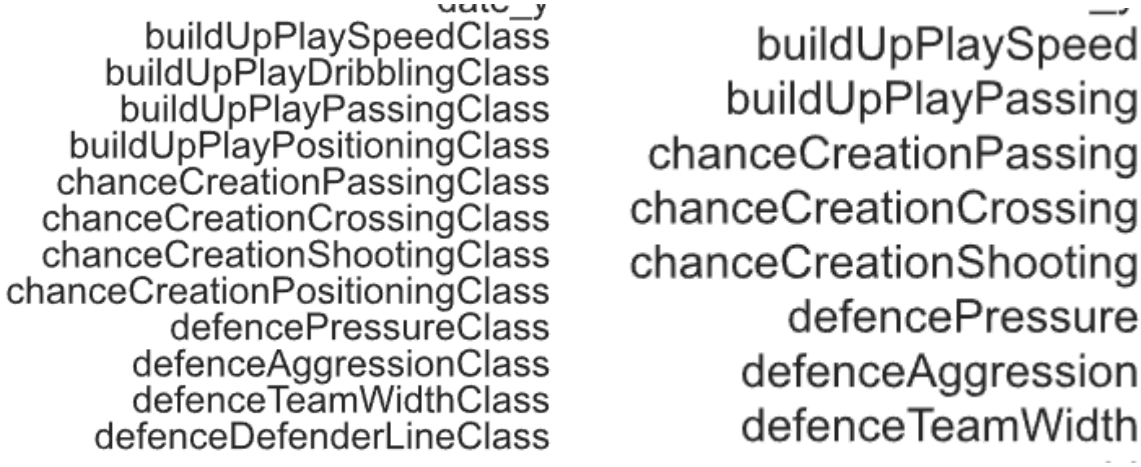
To resolve this issue, the four groups are created based on IQR provided by the EDA. IQR is chosen as it's generated based on median rather than averages.

1) Almost Guaranteed Win (0 – 1.682)
2) Likely Win (1.683 – 2.100)
3) Likely loss(2.101 – 2.684)
4) Likely Loss (2.684 +)

This stratification works as the resulting dataset has a fairly even distribution:



- Next is looking at the independent/input variables.
  i. From the database provided, I will only look at the table marked as "team Attributes" as this is relevant to how the team performs. Individual details are ignored.
  ii. From the initial EDA, I am able to tell the database consists of numerical and categorial independent variables.
  iii. Screenshots below show categorial vars on the left and numerical vars on the left.



iv. For Numerical Data:

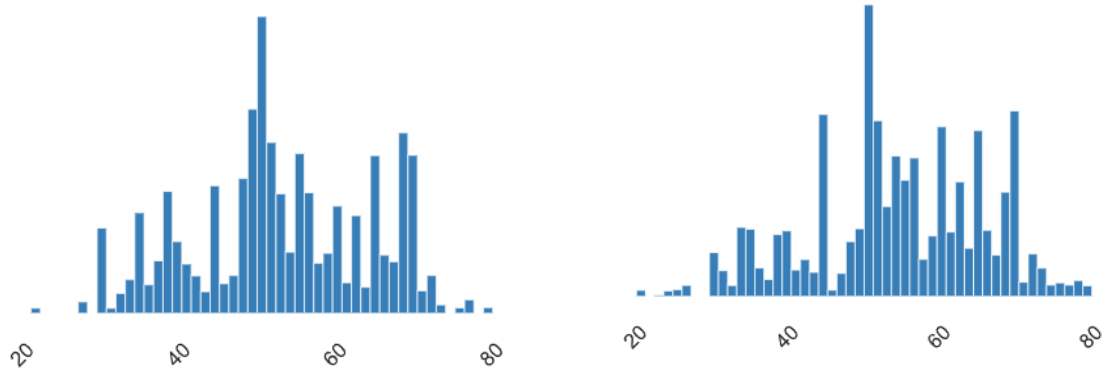
Alerts

`date` has a high cardinality: 1361 distinct values	High cardinality
`id_x` is highly overall correlated with `country_id` and 1 other fields	High correlation
`match_api_id` is highly overall correlated with `season`	High correlation
`B365H` is highly overall correlated with `BWH` and 5 other fields	High correlation
`BWH` is highly overall correlated with `B365H` and 5 other fields	High correlation
`IWH` is highly overall correlated with `B365H` and 5 other fields	High correlation
`LBH` is highly overall correlated with `B365H` and 5 other fields	High correlation
`WHH` is highly overall correlated with `B365H` and 5 other fields	High correlation
`VCH` is highly overall correlated with `B365H` and 5 other fields	High correlation
`avgOdds` is highly overall correlated with `B365H` and 5 other fields	High correlation
`country_id` is highly overall correlated with `id_x` and 1 other fields	High correlation
`league_id` is highly overall correlated with `id_x` and 1 other fields	High correlation
`season` is highly overall correlated with `match_api_id`	High correlation
`homeTeamID` is highly skewed (γ1 = 27.14622012)	Skewed
`away_team_api_id` is highly skewed (γ1 = 21.85942501)	Skewed

The above is the warnings for the EDA report on the numerical variables. None of the warnings are in relations to the predictive input stats. High correlation is showing the betting companies agree with each other the outcomes. Other warnings are mostly included on variables that are used for identification (IDs, dates, names) rather than as an input for prediction.

No outliers are detected. The numbers are on a standardized scale of 0 – 100. Nothing exceeds or goes below this range.

Below are random samples of distributions for the input numerical variables. We can see its mostly normal. With no warnings in EDA report, I can assume normality amongst the inputs.



v. For Categorical Data:

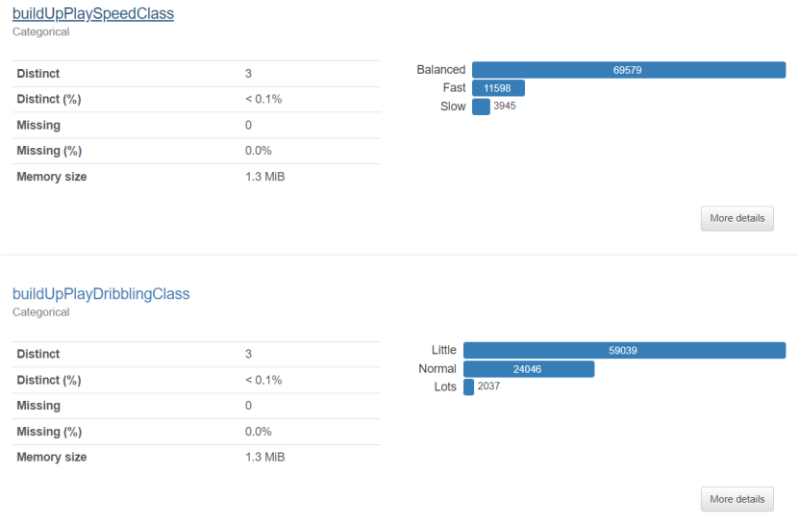
Alerts

`date_x` has a high cardinality: 1361 distinct values	High cardinality
`id_x` is highly overall correlated with `country_id` and 1 other fields	High correlation
`match_api_id` is highly overall correlated with `season`	High correlation
`B365H` is highly overall correlated with `BWH` and 5 other fields	High correlation
`BWH` is highly overall correlated with `B365H` and 5 other fields	High correlation
`IWH` is highly overall correlated with `B365H` and 5 other fields	High correlation
`LBH` is highly overall correlated with `B365H` and 5 other fields	High correlation
`WHH` is highly overall correlated with `B365H` and 5 other fields	High correlation
`VCH` is highly overall correlated with `B365H` and 5 other fields	High correlation
`avgOdds` is highly overall correlated with `B365H` and 5 other fields	High correlation
`country_id` is highly overall correlated with `id_x` and 1 other fields	High correlation
`league_id` is highly overall correlated with `id_x` and 1 other fields	High correlation
`season` is highly overall correlated with `match_api_id`	High correlation
`date_y` is highly overall correlated with `buildUpPlayDribblingClass`	High correlation
`buildUpPlayDribblingClass` is highly overall correlated with `date_y`	High correlation
`buildUpPlayPositioningClass` is highly imbalanced (61.4%)	Imbalance
`defencePressureClass` is highly imbalanced (55.8%)	Imbalance
`defenceAggressionClass` is highly imbalanced (60.7%)	Imbalance
`defenceTeamWidthClass` is highly imbalanced (64.1%)	Imbalance
`defenceDefenderLineClass` is highly imbalanced (57.3%)	Imbalance
`homeTeamID` is highly skewed (γ1 = 27.14622012)	Skewed
`away_team_api_id` is highly skewed (γ1 = 21.85942501)	Skewed

The above is the warnings for the EDA report on the categorical variables. (Note the report included other items such as date, names, IDs, as well).

Again, no outliers are detected.

All previous issues with the identification variables (IDs, etc.) remain true. However, this time there is a great imbalance in distribution for many of the variables. Showing two randomly picked examples below, normality is not a given:



This categorical trend does show one striking piece of information. There is a lot of agreement between different teams on they play. It seems most teams play one way and there's not a of variation.

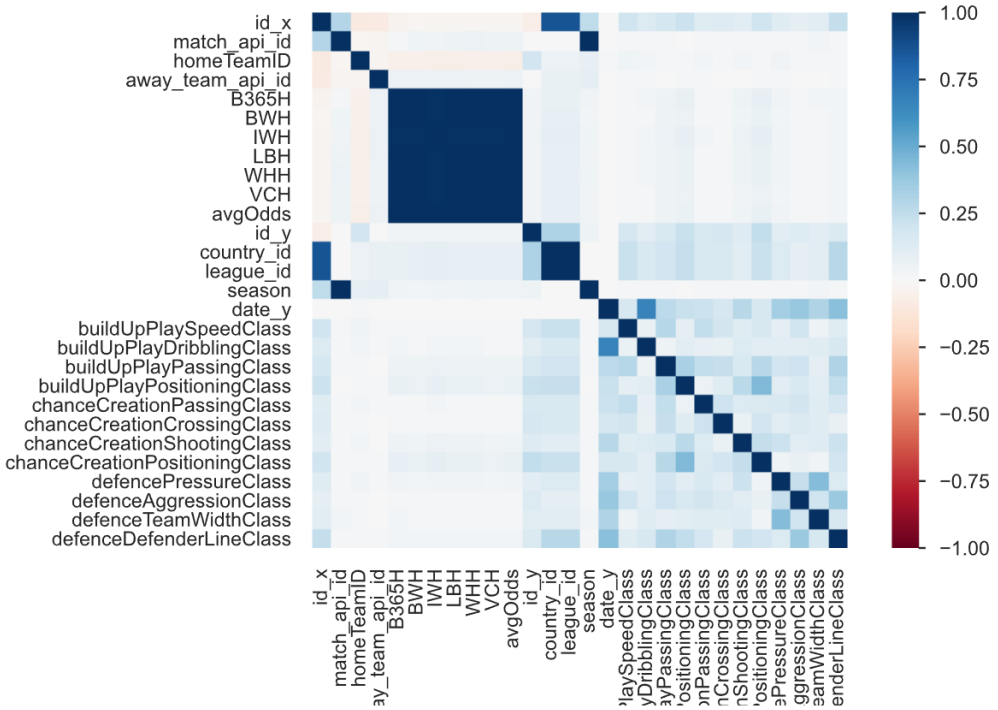
Shapiro Wilkes performed on output statistics avgOdds. It's safe to say this is not a normal distribution. As such non-parametric tests should be used.

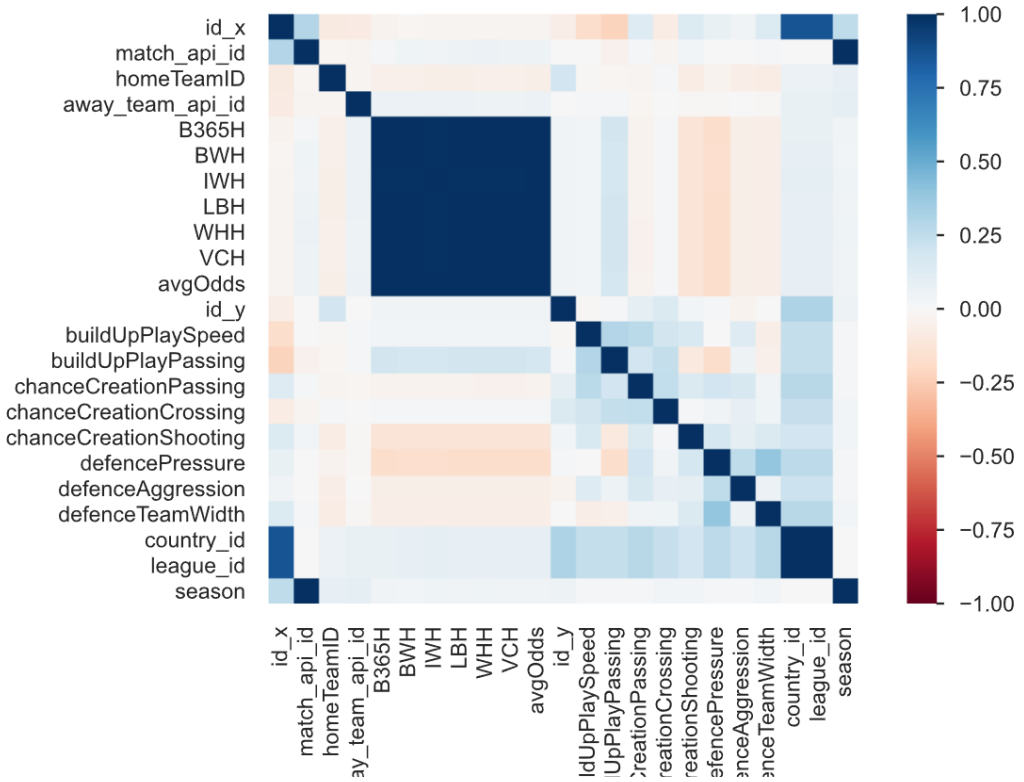
```
ShapiroResult(statistic=0.6385138034820557, pvalue=0.0)
```

Normalization is not done on the numerical data as the data is all already on a scale of 0 – 100. As for the Categorical data, an amateur, hard coded encoding was done to covert the strings into a workable set of ordinal data. Each class was converted into a scale of 1 – 3 (depending on the number of categories) with the lower end (1) always representing a slower, restrictive, defensive style of play and the higher end (3) representing a fast-flowing, positive attacking style of play.

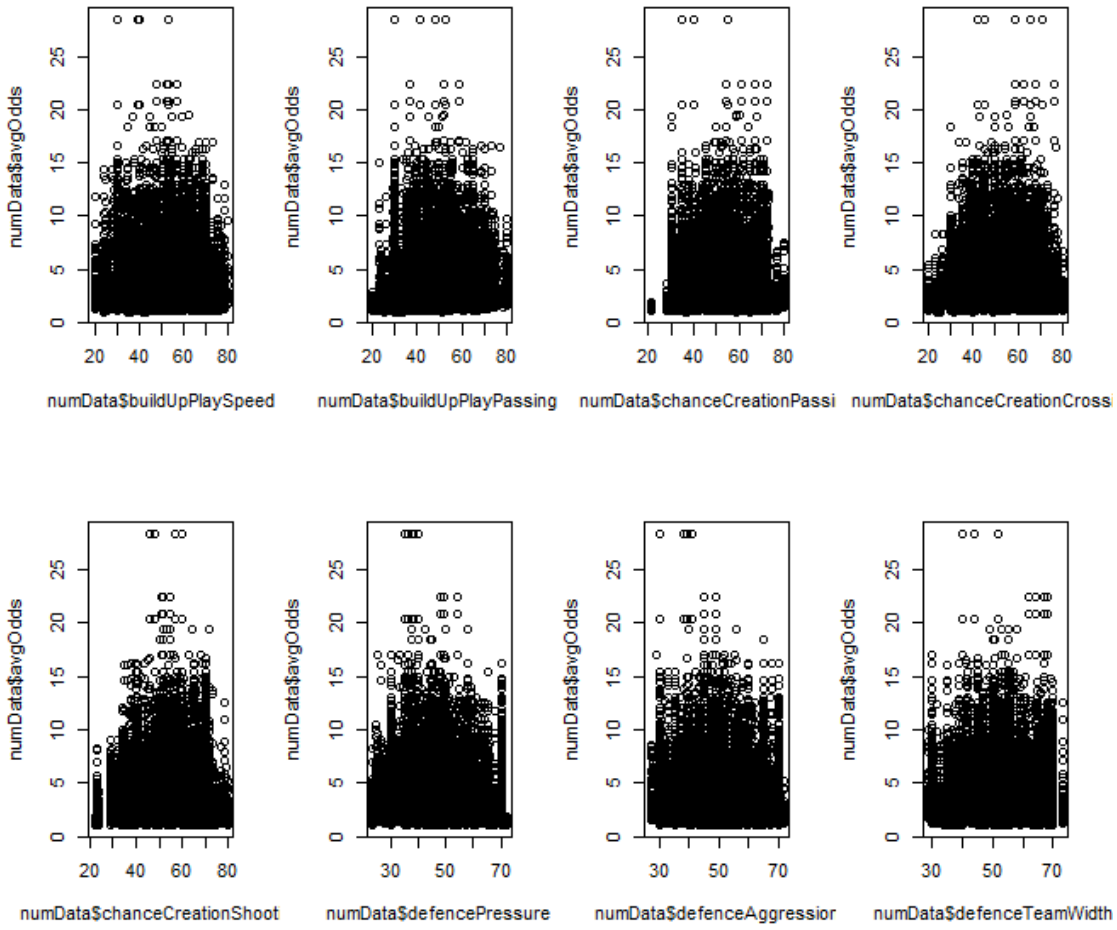
As for multicollinearity, below are the two charts for the correlations between all variables on both the numerical dataset and the categorial. As previously mentioned. The only high correlation belongs to the betting odds (which indicate agreement, to deal with this an average is taken), the irrelevant variables (id, teamname, date, etc.). The variables that are to be the input, do not show any hint of high levels of positive nor negative correlations amongst themselves. At least, not to the extend where the EDA reported needed to warn me.

Having looked at the various ‘interactions’ between input variables on the EDA reports, there also seems to show little to no correlation between the relevant inputs themselves.

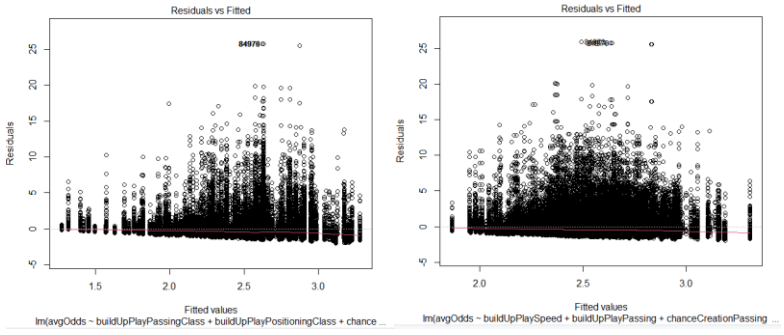




Below are the scatterplots created for the numerical data. From the quick observation below, it's safe to say linearity cannot be assumed for the numerical data.



Next set of charts show the fitted values vs residuals and this can tell me about the assumption of homoscedasticity. The left chart shows the categorical data and the spread would suggest heteroscedasticity. The right chart shows numerical data and the spread would also suggest heteroscedasticity.



- Now that an understanding of the independent variables is completed. I will attempt to combine the two into one dataframe.

- The categorical data is crudely encoding using the following way:
    - Each category is looked at and converted into a scale of 1 – 3 or 1 – 2 depending on the number of categories.
    - 1 and lower numbers are associated with a restrictive, slow, defensive style of play. 3 and higher number are associated with a faster, free-flowing attacking style of play.
- All values were put onto one dataframe and a new EDA report is generated. A couple of key things to note as shown by the warnings below:

`buildUpPlaySpeedClass` has 28133 (33.8%) missing values

`buildUpPlayDribblingClass` has 28133 (33.8%) missing values

`buildUpPlayPassingClass` has 28133 (33.8%) missing values

`buildUpPlayPositioningClass` has 28133 (33.8%) missing values

`chanceCreationPassingClass` has 28133 (33.8%) missing values

`chanceCreationCrossingClass` has 28133 (33.8%) missing values

`chanceCreationShootingClass` has 28133 (33.8%) missing values

`chanceCreationPositioningClass` has 28133 (33.8%) missing values

`defencePressureClass` has 28133 (33.8%) missing values

`defenceAggressionClass` has 28133 (33.8%) missing values

`defenceTeamWidthClass` has 28133 (33.8%) missing values

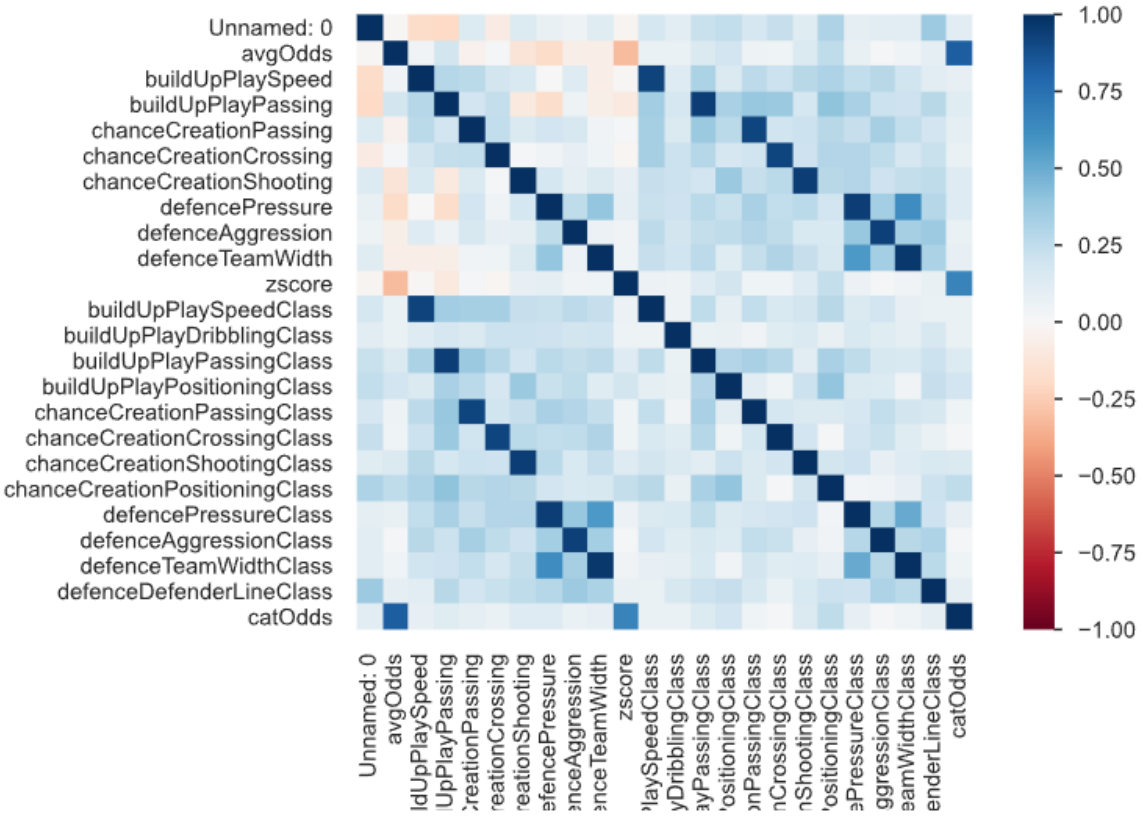
`defenceDefenderLineClass` has 28133 (33.8%) missing values

- Categorical data has a good chunk of missing data throughout.

`buildUpPlaySpeed` is highly overall correlated with `buildUpPlaySpeedClass`	High correlation
`buildUpPlayPassing` is highly overall correlated with `buildUpPlayPassingClass`	High correlation
`chanceCreationPassing` is highly overall correlated with `chanceCreationPassingClass`	High correlation
`chanceCreationCrossing` is highly overall correlated with `chanceCreationCrossingClass`	High correlation
`chanceCreationShooting` is highly overall correlated with `chanceCreationShootingClass`	High correlation
`defencePressure` is highly overall correlated with `defencePressureClass` and 1 other fields	High correlation
`defenceAggression` is highly overall correlated with `defenceAggressionClass`	High correlation
`defenceTeamWidth` is highly overall correlated with `defencePressureClass` and 1 other fields	High correlation
`zscore` is highly overall correlated with `catOdds`	High correlation
`buildUpPlaySpeedClass` is highly overall correlated with `buildUpPlaySpeed`	High correlation
`buildUpPlayPassingClass` is highly overall correlated with `buildUpPlayPassing`	High correlation
`chanceCreationPassingClass` is highly overall correlated with `chanceCreationPassing`	High correlation
`chanceCreationCrossingClass` is highly overall correlated with `chanceCreationCrossing`	High correlation
`chanceCreationShootingClass` is highly overall correlated with `chanceCreationShooting`	High correlation
`defencePressureClass` is highly overall correlated with `defencePressure` and 2 other fields	High correlation
`defenceAggressionClass` is highly overall correlated with `defenceAggression`	High correlation
`defenceTeamWidthClass` is highly overall correlated with `defencePressure` and 2 other fields	High correlation

- Categorical data is very highly correlated with a lot of the numerical data. This is extremely evident in the correlations matrix produced by the EDA (shown below). From here it seems only 3 of the categorical variables are not correlated:
    - buildUpPlayDribblingClass
    - chanceCreationPositionClass
    - defenceDefenderLineClass
- Due to the high correlations between most of the categorical data and the numerical data, the whole of the categorical data frame will be simply dropped.
    - Upon a further look at what's actually included in categorical data – it looks like they (the database owners) simply applied a classification on existing numerical data. The categorial data and numerical data (that are correlated) are simply different versions of themselves.
    - Due to this fact, the majority of the categorical dataframe can be considered redundant.

- The 33.8% missing data is also too significant to simply take an average or ignored. The 3 non-correlated variables will also be dropped due to their significantly high missing values %.



After all the considerations above, the dataframe that is left over is simply just the numerical data.

2) Feature Selection

Given:

Output/Independent – Categorical, even distribution.

Input/Dependent – Numerical, normal.

- <u>Filter Based – Information Gain:</u>

The first feature selection method applied is using information gain to filter out the best.

Using a rank and sort filter, below shows the numerical variables with the highest information gain:

defencePressure	0.052977
chanceCreationPassing	0.051994
buildUpPlayPassing	0.051350
chanceCreationShooting	0.049089
chanceCreationCrossing	0.037899
buildUpPlaySpeed	0.034756
defenceTeamWidth	0.031156
defenceAggression	0.023384

From the details above, we can see first and foremost, none of the variables are super strong objectively as they all rank 0.05 or lower. This tells me the overall task to trying to predict victory in a football match is hard, it's not going to be the most accurate prediction. It helps tell me as a baseline fact that football is simply unpredictable, and hence any factors for prediction of results should be taken with a grain of salt.

Relatively speaking, defencePressure, chanceCreatePassing, and buildUpPlayPassing seem to contribute the most to an accurate prediction of victory.

The efficiency of this feature selection shows a runtime of 3.15 secs..

- Wrapper Based – Backwards Selection

The first thing that must be done is to change the categorical output variable into a number. I've tried multiple times on different machines to run a backwards selection in R using the string variable "catOdds" as the independent variable. All of which produces an error saying "Na/NaN/Inf in'y'". I ran checks to see if there are any NaN or Inf or Na in the column but there are none. Based on this, I'm assuming lm() simply just doesn't take strings as the independent variable. This is the reason why conversion of the categorical output data must be done.

As such, I will convert the strings into numbers. I will do this extremely simply by converting the strings into a ranked set of numbers:

1 = Almost Guaranteed Win

2 = Likely Win

3 = Unlikely Win

4 = Likely Loss

Using this, I will treat the numbers like variables. The model will simply tell about how each variable can help predict 1, 2, 3 or 4.

The following are the results generated in R:

```
Call:
lm(formula = catOddsNUm ~ buildUpPlaySpeed + buildUpPlayPassing +
    chanceCreationPassing + chanceCreationCrossing + chanceCreationShooting +
    defencePressure + defenceAggression + defenceTeamWidth, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-2.38228 -0.92251 -0.07763  0.84191  2.26063

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             2.9268354  0.0392165  74.633  < 2e-16 ***
buildUpPlaySpeed        0.0014051  0.0003547   3.962 7.44e-05 ***
buildUpPlayPassing      0.0154454  0.0003691  41.844  < 2e-16 ***
chanceCreationPassing  -0.0040099  0.0003680 -10.898  < 2e-16 ***
chanceCreationCrossing -0.0013828  0.0003461  -3.995 6.47e-05 ***
chanceCreationShooting -0.0073509  0.0003522 -20.872  < 2e-16 ***
defencePressure        -0.0117128  0.0004277 -27.388  < 2e-16 ***
defenceAggression      -0.0025587  0.0004049  -6.320 2.63e-10 ***
defenceTeamWidth        0.0016611  0.0004476   3.711 0.000207 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.074 on 83131 degrees of freedom
Multiple R-squared:  0.05824,   Adjusted R-squared:  0.05815
F-statistic: 642.7 on 8 and 83131 DF,  p-value: < 2.2e-16
```

```
Start:  AIC=11876.22
catOddsNUm ~ buildUpPlaySpeed + buildUpPlayPassing + chanceCreationPassing +
    chanceCreationCrossing + chanceCreationShooting + defencePressure +
    defenceAggression + defenceTeamWidth

                        Df Sum of Sq   RSS   AIC
<none>                              95886 11876
- defenceTeamWidth       1     15.89 95901 11888
- buildUpPlaySpeed       1     18.11 95904 11890
- chanceCreationCrossing 1     18.41 95904 11890
- defenceAggression      1     46.07 95932 11914
- chanceCreationPassing  1    136.98 96023 11993
- chanceCreationShooting 1    502.46 96388 12309
- defencePressure        1    865.19 96751 12621
- buildUpPlayPassing     1   2019.53 97905 13607
```

Looking at the AIC generated (given low AIC = lower amounts of information loss from removal the respective variable), the amount of information loss from each variable is fairly similar.

Generally, speaking, the lower the AIC the better the model. This model has a pretty high one, as such it's not the best.

This also failed to eliminate any variables. This feature selection will be taken with a grain of salt.

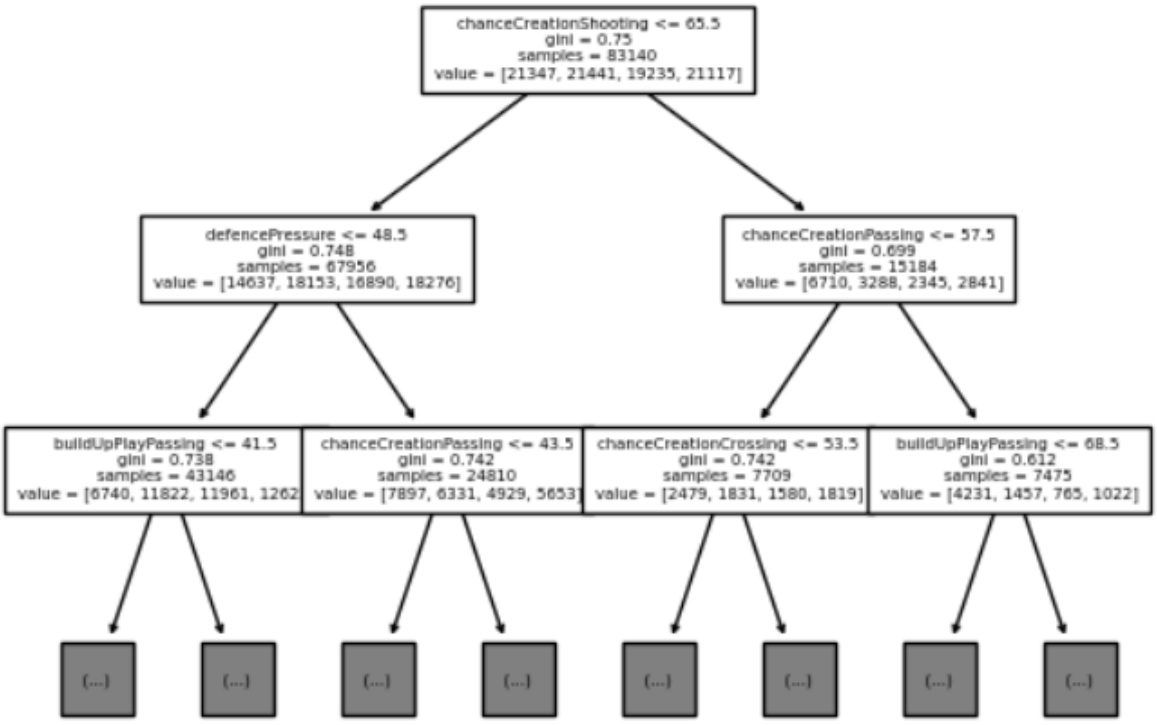
The Adjusted R-squared value also shows a very low amount of relationship.

The backwards selection here has told me the victory results are not easily predicted using the input variables. This does make sense, as if it's easy to predict, sports betting would be a poor business.

- Embedded Method – Decision Tree

Final method deployed is the embedded method where feature selection is a part of the model building.

A decision tree was created on the input dataset, and generated the following abridged tree:



The full tree is not displayed as the full details are irrelevant to the scope of this project.

The decision tree is built based on GINI impurity splitting, which aims to create the purest nodes possible in each of the splits. Unfortunately, if we do expand the tree and look at the details, GINI does not get low. A copy of the full text version of decision tree is attached in

the Github repository but GINI doesn't get lower than roughly 0.6. This tell me it's not the best and most accuracy tree ever. However, that's not the point of this project, the point is to look at the influential input variables, which sit on top of the tree.

The key takeaway here is that only 1 of which is related to defence.

- Bonus method – SelectKBest

Using the recommended package, I will use this function to try and do feature selection as well.

I will be using K select with respect to the Chi Squared test, as this is an appropriate test to use when the output variable is categorical.

The k will set as 3 as I want to simply see the top 3 best predictors.

From the results of my k select, I am able to see the following are the top 3 predictors:

1. chanceCreationPassing
2. chanceCreationShooting
3. defencePressure

Interestingly, only 1 of 3 top features includes a defensive one.

Extending the "k", I can see the following being top influencers as well. More defensive stats are included:

4. buildUpPlayPassing
5. defenceTeamWidth
6. defenceAggression

3) Model Building

Here, I propose to build 3 different classification models to predict high chances of winning. Each model will include a baseline model that includes all the input variables, and a defense model that includes only the defense variables, and a comparison model that includes all variables except defensive ones. In total there will be 9 models.

The accuracies and various relevant statistics between the baseline and defence models in order to generate knowledge about just how important defence is to winning.

The 3 models that will be built are:

- Decision Tree
- Random Forest

- KNN Classifier

We'll defined defensive styles as having the following independent variables:

- defencePressure
- defenceAggression
- defenceTeamWidth

As such, non-defensive is simply the baseline with these variables removed.

In terms of **stability,** the following measures were done. For the building if all models, we will always use a 70:30 training to testing ratio with a random state of 88 for consistency amongst all models.

The cross-validation technique used to train all models in this case is simply the holdout method where the dataset was simply split into training and testing. There are stronger cross validation techniques such as k-fold, but due to resources limitations (particularly that of time and ability) the researcher has chosen to keep things simple. For the scope of the question, we are simply looking to compare the relative accuracies of each generated model. As long as the cross-validation technique remains consistent and all the model building remains consistent, we can make a good comparison. Strength and ability of the model is not as important as consistency since the goal is simply comparison.

Please note efficiency calculations are taken on an average basis. The exact millisecond value of each run varies. The values recorded represent an average and should be treated as "the value" +/- a couple of milliseconds.

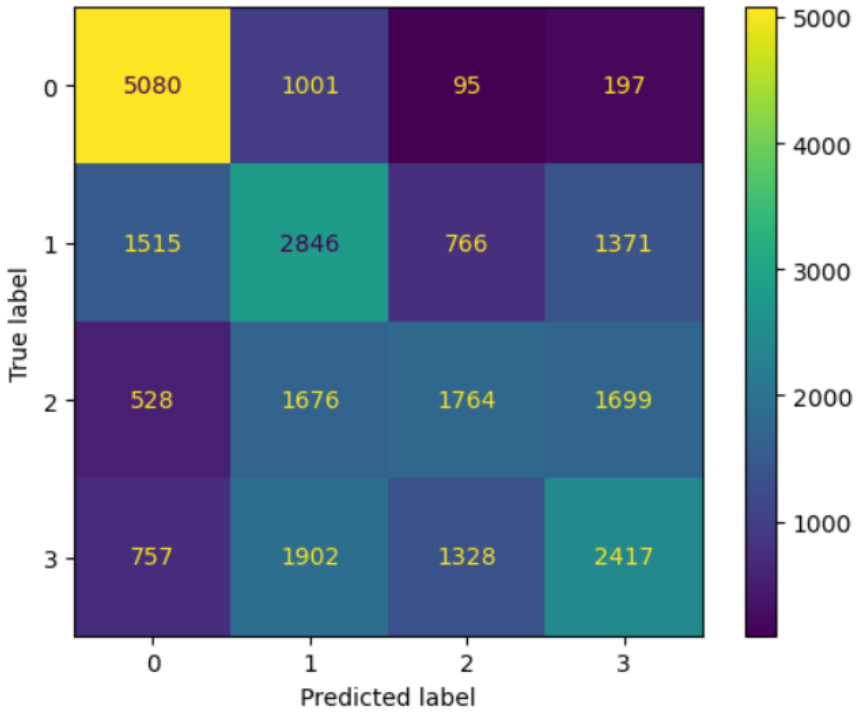
Decision Tree:

Before proceeding, please see notes in Appendix B. Due to the sheer size of the full tree, and the scope of the question of this project, I am only looking at the key indicator variables, which are on the top of the decision tree. This applies for all three iterations of the decision tree.

All trees are pruned to only show the top 3 layers of the tree as these are the features we are interested in for the scope of this project (strongest influencers). The trees generated are very massive and wide, most of it is trimmed so we can show the key features elegantly. See Appendix B for more details on the full tree.

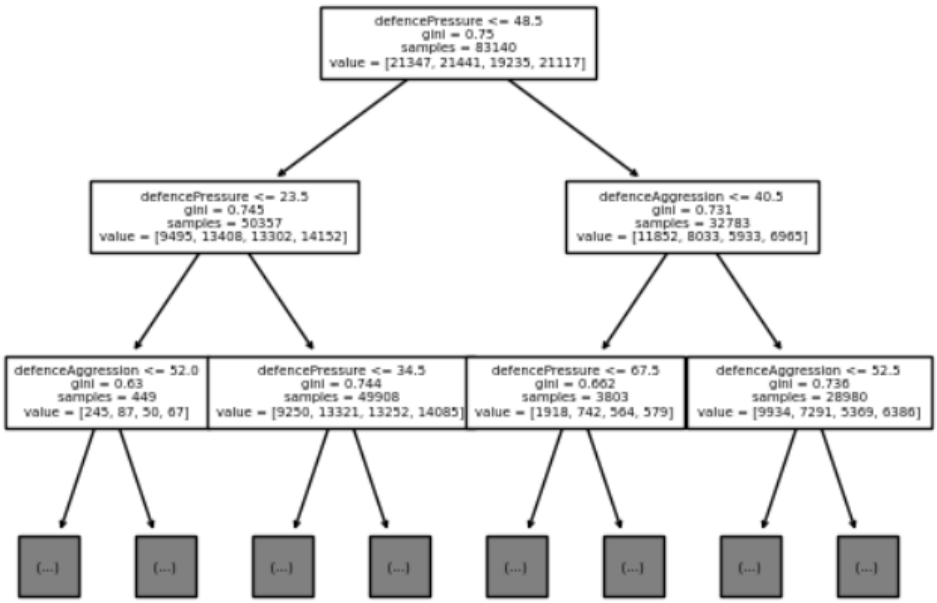
Baseline decision tree model was created during the embedded feature selection.

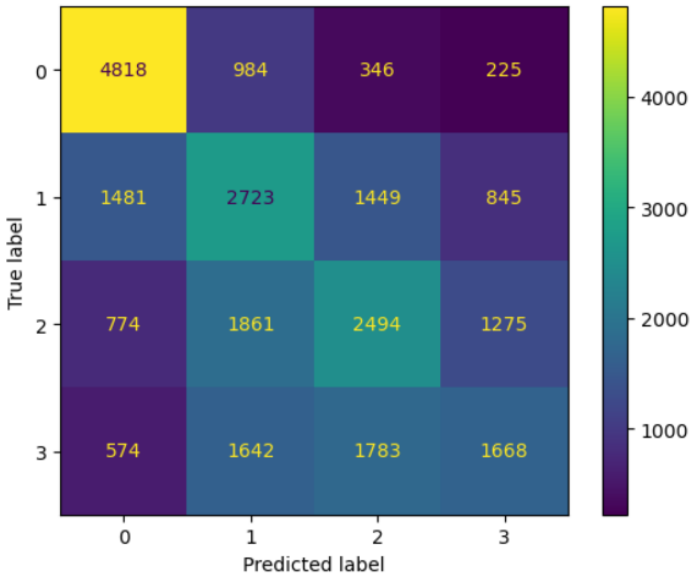
Below is the confusion matrix and confusion report for the baseline:



	precision	recall	f1-score	support
Almost Guaranteed Win	0.64	0.80	0.71	6373
Likely win	0.38	0.44	0.41	6498
Likley loss	0.45	0.31	0.37	5667
Unlikely win	0.43	0.38	0.40	6404
accuracy			0.49	24942
macro avg	0.47	0.48	0.47	24942
weighted avg	0.48	0.49	0.47	24942

A decision tree that includes only the defensive variables are predictors is generated, and the following abridged tree is a quick show of the classification being done on it:

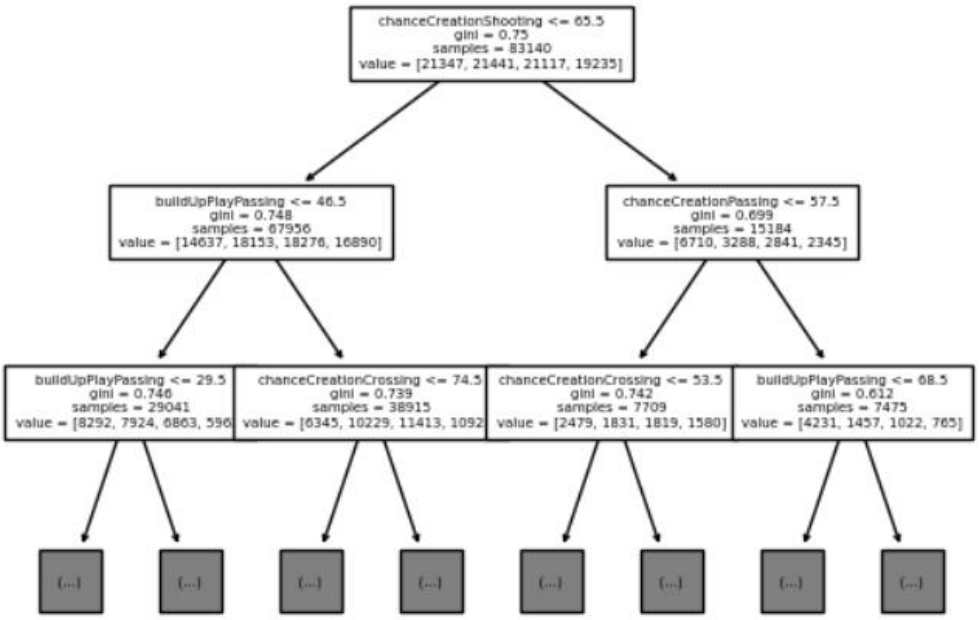

defencePressure <= 48.5
gini = 0.75
samples = 83140
value = [21347, 21441, 19235, 21117]
defencePressure <= 23.5
gini = 0.745
samples = 50357
value = [9495, 13408, 13302, 14152]
defenceAggression <= 40.5
gini = 0.731
samples = 32783
value = [11852, 8033, 5933, 6965]
defenceAggression <= 52.0
gini = 0.63
samples = 449
value = [245, 87, 50, 67]
defencePressure <= 34.5
gini = 0.744
samples = 49908
value = [9250, 13321, 13252, 14085]
defencePressure <= 67.5
gini = 0.662
samples = 3803
value = [1918, 742, 564, 579]
defenceAggression <= 52.5
gini = 0.736
samples = 28980
value = [9934, 7291, 5369, 6386]
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)

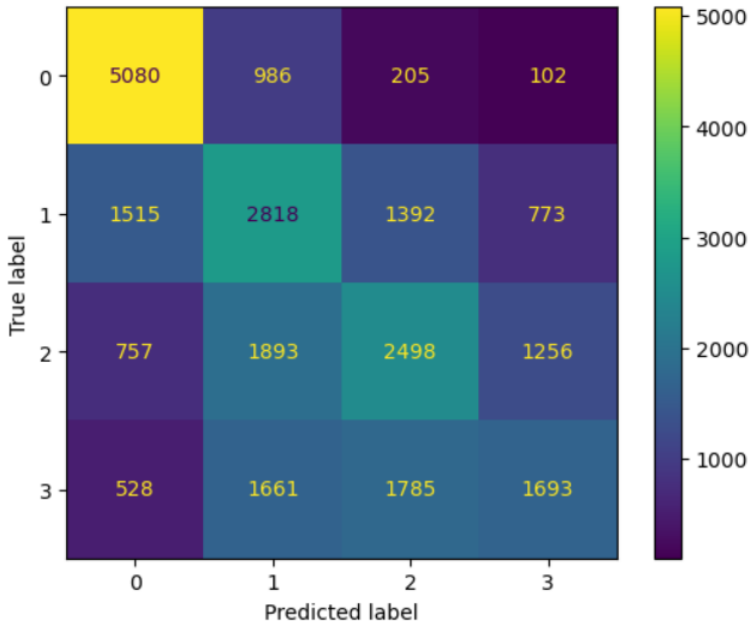

4818
984
346
225
1481
2723
1449
845
774
1861
2494
1275
574
1642
1783
1668
0
1
2
3
True label
0
1
2
3
Predicted label
4000
3000
2000
1000

	precision	recall	f1-score	support
1	0.63	0.76	0.69	6373
2	0.38	0.42	0.40	6498
3	0.41	0.39	0.40	6404
4	0.42	0.29	0.34	5667
accuracy			0.47	24942
macro avg	0.46	0.46	0.46	24942
weighted avg	0.46	0.47	0.46	24942

The tree goes down many levels and gets very specific with the sorting. Notice how defencePressure is on the top level as well as the second level.

The following is the abridge diagram to the "non-defensive" model aka the model that has defensive variables removed:





	precision	recall	f1-score	support
1	0.64	0.80	0.71	6373
2	0.38	0.43	0.41	6498
3	0.42	0.39	0.41	6404
4	0.44	0.30	0.36	5667
accuracy			0.48	24942
macro avg	0.47	0.48	0.47	24942
weighted avg	0.47	0.48	0.47	24942

By generating the accuracy scores, I am able to see the following as a comparison:

Model	Accuracy Score	Efficiency (secs)
Defensive	0.469	1.19
Non-Defensive	0.485	1.26
Baseline	0.485	0.49

First things first, all models produce not great accuracies. Each of which produce a rough accuracy of 46 – 48%. This does make sense is a real world application as if I am able to create a model that can accurately predict betting odds, I would be a billionaire and invest all my money into sports betting. This just tells me football wins are not easily nor accurately predicted.

As mentioned in the previous sections, the high gini values do confirm the accuracies of this model wouldn't be the best.

Whatever difference exists is very minor.

Interestingly, the purely defensive model shows the lowest accuracy score. The baseline model as well as the non-defensive model both show higher accuracies shows with very little difference in between them.

The decision tree model shows me that defense is not the key predictor to victory. It has a say, but its voice is essentially unheard. It does not have a strong influence on the dependent variable.

Random Forest:

For Random Forest, a similar numerical variable is imposed on the dataset like we had done on backwards selection above. Rather than using the string variables as the dependent variable, I used their numerical counterpart:

1 = Almost Guaranteed Win

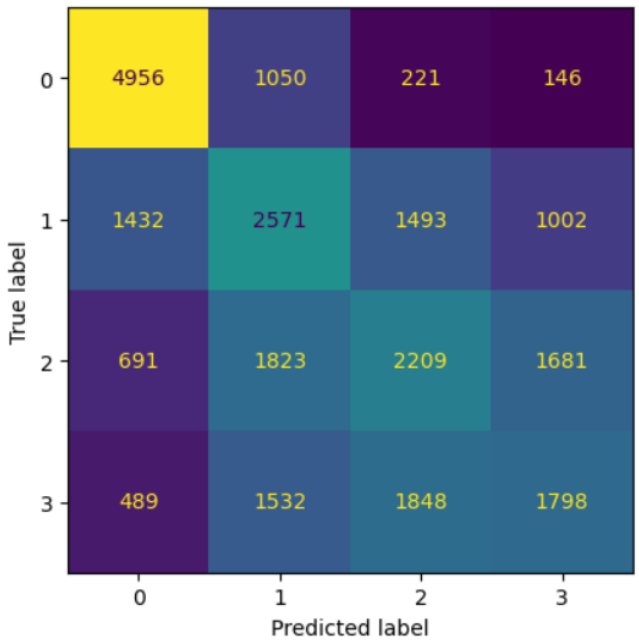
2 = Likely Win

3 = Unlikely Win

4 = Likely Loss

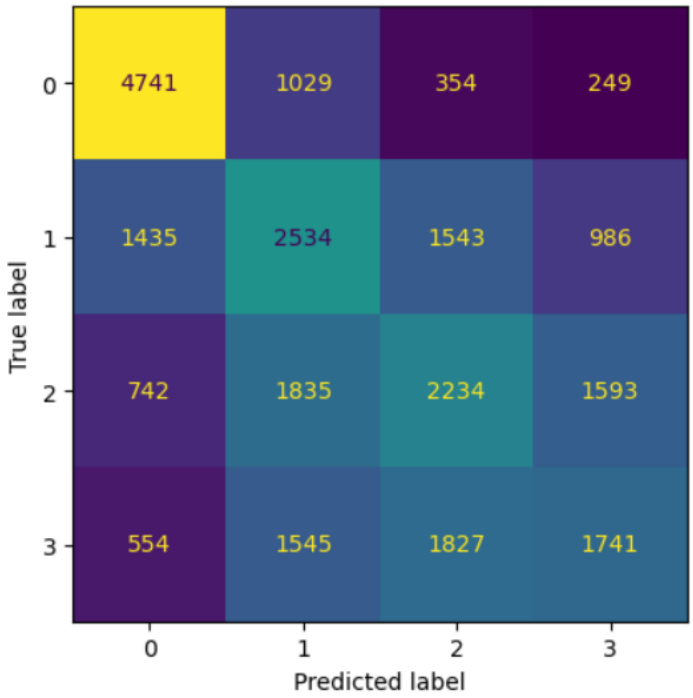
To keeps things consistent, the default values for the classifiers are used. Meaning we are generating 100 trees in the forest. The criterion used to measure quality of a split is GINI (just like decision tree model). There was no pruning done.

Below is the confusion matrix and classification report for the baseline model:



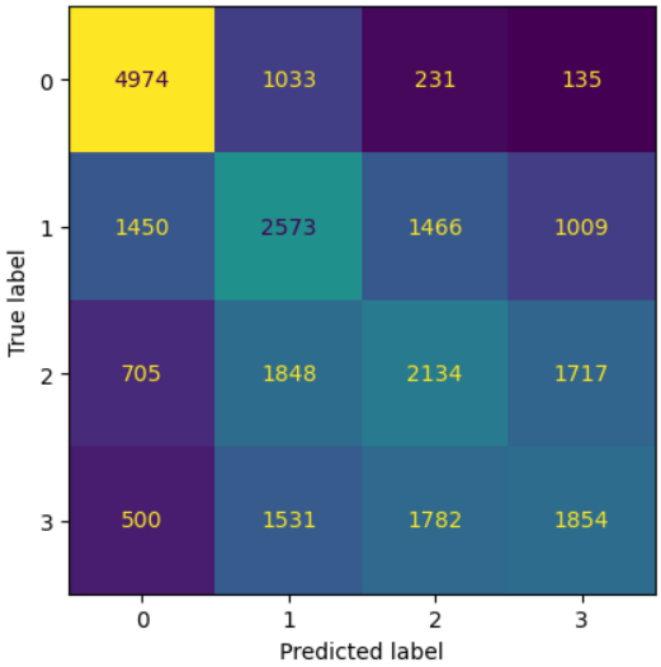
	precision	recall	f1-score	support
1	0.65	0.78	0.71	6373
2	0.37	0.41	0.39	6498
3	0.38	0.33	0.35	6404
4	0.39	0.32	0.35	5667
accuracy			0.46	24942
macro avg	0.45	0.46	0.45	24942
weighted avg	0.45	0.46	0.45	24942

Below is the confusion matrix and classification report for the defensive model:



	precision	recall	f1-score	support
1	0.64	0.74	0.69	6373
2	0.36	0.39	0.38	6498
3	0.38	0.35	0.36	6404
4	0.38	0.31	0.34	5667
accuracy			0.45	24942
macro avg	0.44	0.45	0.44	24942
weighted avg	0.44	0.45	0.44	24942

Below is the confusion matrix and classification report for the non-defensive model:



	precision	recall	f1-score	support
1	0.65	0.78	0.71	6373
2	0.37	0.39	0.38	6498
3	0.38	0.33	0.36	6404
4	0.39	0.33	0.36	5667
accuracy			0.46	24942
macro avg	0.45	0.46	0.45	24942
weighted avg	0.45	0.46	0.45	24942

Below is a chart to summarize the respective accuracies of each model under a random forest classifier:

Model	Accuracy Score	Efficiency (secs)
Defensive	0.4510	2.18
Non-Defensive	0.4625	3.27
Baseline	0.4624	3.18

Overall, this model produced a similar accuracy score. However, an average accuracy of roughly 46% is still not good. Not enough for me to quit my job and do sports betting full time using my algorithm. Using this algorithm, I'm still most likely to lose any bets I place.

Looking at the comparisons, once again the difference in accuracies is minute. The defensive model this time created the, relatively, lowest accuracy with 45%. The non-defensive model showed a tiny improvement (0.001%) from the baseline model. These scores suggest by

removing defensive, the predictions will be better by 0.0001%. Honestly, an improvement of 0.0001% in this context is near negligible.

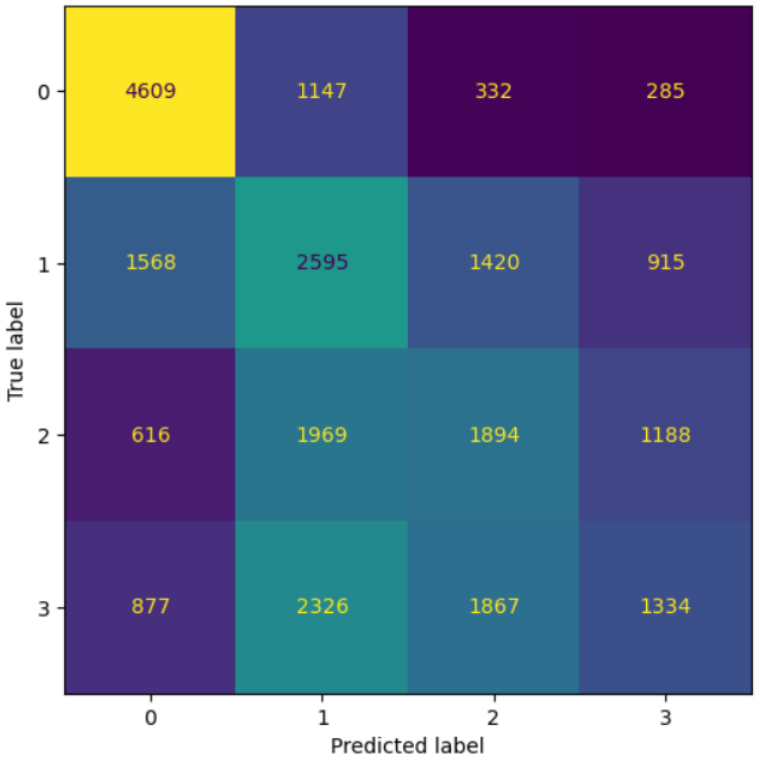
KNN Classifier

Finally, the KNN classifier models are created.

For these comparisons keep consistent a number of neighbours value “k” was set as 5 for all models.

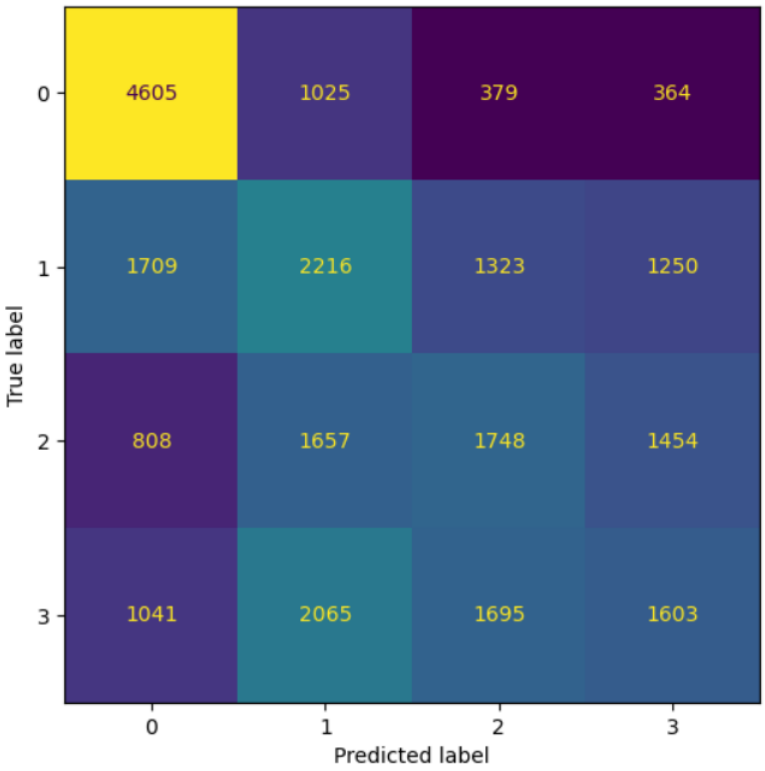
Below are the respective confusion matrices and classification reports of the various KNN models:

Baseline:



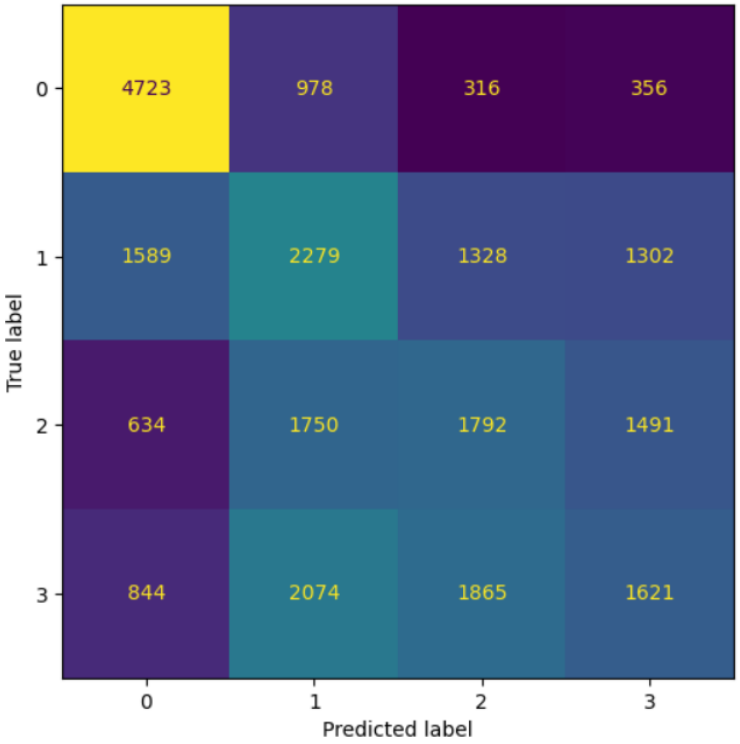
	precision	recall	f1-score	support
Almost Guaranteed Win	0.60	0.72	0.66	6373
Likely win	0.32	0.40	0.36	6498
Likley loss	0.34	0.33	0.34	5667
Unlikely win	0.36	0.21	0.26	6404
accuracy			0.42	24942
macro avg	0.41	0.42	0.40	24942
weighted avg	0.41	0.42	0.41	24942

Defensive:



	precision	recall	f1-score	support
Almost Guaranteed Win	0.56	0.72	0.63	6373
Likely win	0.32	0.34	0.33	6498
Likley loss	0.34	0.31	0.32	5667
Unlikely win	0.34	0.25	0.29	6404
accuracy			0.41	24942
macro avg	0.39	0.41	0.39	24942
weighted avg	0.39	0.41	0.40	24942

Non-Defensive:



	precision	recall	f1-score	support
Almost Guaranteed Win	0.61	0.74	0.67	6373
Likely win	0.32	0.35	0.34	6498
Likley loss	0.34	0.32	0.33	5667
Unlikely win	0.34	0.25	0.29	6404
accuracy			0.42	24942
macro avg	0.40	0.42	0.40	24942
weighted avg	0.40	0.42	0.41	24942

The following chart comparing accuracies is derived:

Model	Accuracy Score	Efficiency (secs)
Defensive	0.4078	1.49
Non-Defensive	0.4176	1.56
Baseline	0.4183	1.67

Similar to all models above, accuracy is not strong. This is still not favorable and reliable to make predictions on betting odds.

Similar to previous models, the defensive model's accuracy predictions seem to be the relative lowest. Telling me these factors would have the less of a say in the overall prediction of a winning result.

Unlike previous however, the baseline model here seem to perform stronger than the model that has no defensive stats in it.

4) Analysis

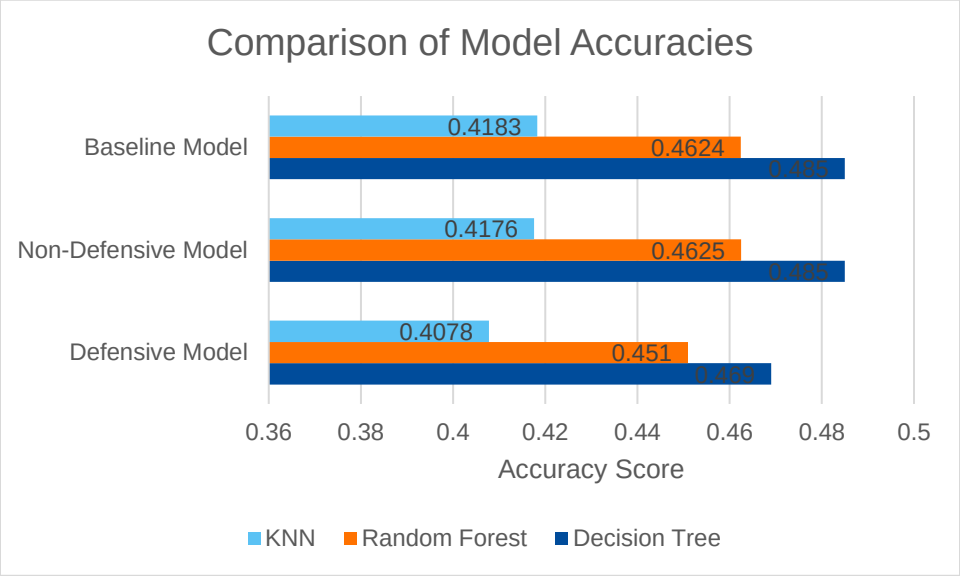
Before starting with the analysis, the following will be defined for the scope of this project:

Term	Definition in the Scope of This Project
Accuracy	= # of correct predictions / total # of predictions
Positive/Negative Outcome	(Positive is) The prediction class in question, relative to what we want. E.g. If want to look for Guaranteed Wins, then this is the positive outcome, all other outcomes are considered negative.
True Positive (TP)	Outcome where the model correctly predicts the positive outcome.
False Positive (FP)	Outcome where the model incorrectly predicts the positive outcome.
True Negative (TN)	Outcome where the model correctly predicts the negative outcome.
False Negative (FN)	Outcome where the model incorrectly predicts the negative outcome.
Precision	What proportion of “positive” classification was actually correct. = TP / (TP + FP) Tells us what % of the prediction for the positive outcome is correct. E.g. if Precision for “Likely Win” = 50% then, when the model predicts a point to “Likely Win”, it’s correct 50% of the time.
Recall	What proportion of actual positives was identified correctly, = TP / (TP + FN) Tells us how much of an outcome class if correctly identified.

	E.g. if Recall for "Likely Win" is 75%, then it correctly identifies a point as "Likely Win" 75% of the time.
F1-Score	Harmonic mean of the precision and recall. It's a measure of the model's accuracy that's able to capture the nuances of the different types of error that may occur. = TP / (TP + 1/2[FP + FN])
Support	Number of "TRUE" responses that lie in each of the class variables. Number of points that's classified as each output variable.

Below is a summary of all the various **accuracies** generated from each model:

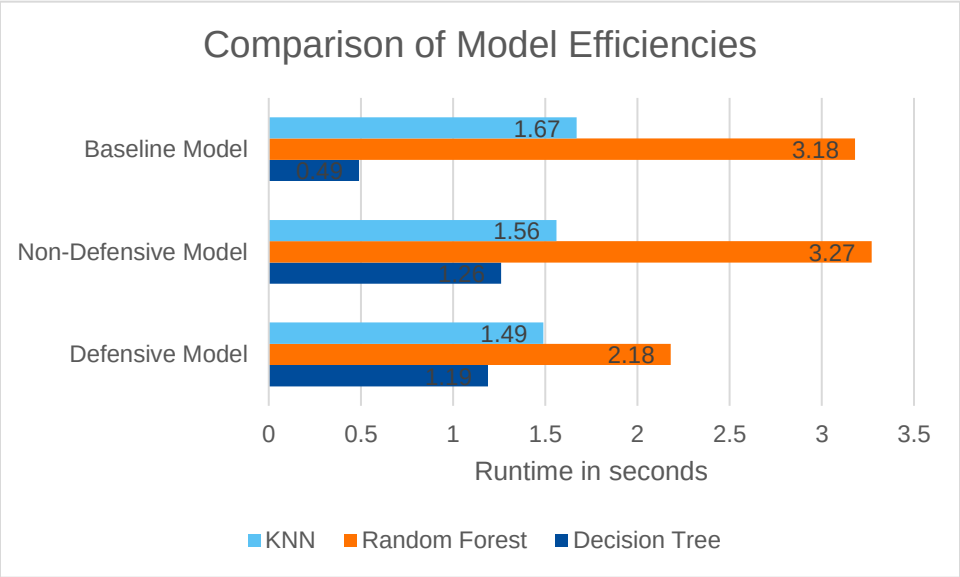
	Decision Tree	**Random Forest**	**KNN**
Defensive Model	0.469	0.4510	0.4078
Non-Defensive Model	0.485	0.4625	0.4176
Baseline Model	0.485	0.4624	0.4183



Below is a summary of all the various **efficiencies** generated from each model:

	Decision Tree	**Random Forest**	**KNN**
Defensive Model	1.19	2.18	1.49
Non-Defensive Model	1.26	3.27	1.56

Baseline Model	0.49	3.18	1.67



The goal of this project was to analyze if a defensive style of play can lead to success. We've defined success as having more wins and generally having betting odds that indicate you're the favorable winner.

As such the efficiencies of the models are good to know but not relevant to the scope of the question. It's interesting to note however, the decision tree would be the optimal model to use as it provided the (relatively) highest accuracy and lowest efficiency (good in my case as it means lowest runtimes) measures.

We've defined defensive styles as having the following independent variables:

- defencePressure
- defenceAggression
- defenceTeamWidth

The aim was to see if models with these variables included or model with solely these variables will generate relatively better prediction accuracies.

Based on the various models above, all of the defensive models seem to create a lower relative level of accuracy. There tend to be a very small difference between the baseline tree and the model where defensive attributes were removed. This can also be seen in the details provided by backwards selection. It showed a fairly high AIC alongside a low adjusted R-squared telling me these input variables do influence, but it's not the strongest nor the most related. This says despite the relationship between the input and the output, any sole focus on any of the input variables is not a guaranteed result of victory. In fact, it influences the victory in a very minute way.

This tells me defensive is not the sole predictor. The accuracies and information above tell me defense is important to a team's success, but it cannot succeed by itself. A mix of all parts of the game is needed to help lead to a team getting better winning odds.

Looking at all the decision trees build, one of the strongest influencers is defense pressure, but the rest of the strong influencers are not defence based with more chance creation variables mentioned. Once again, this confirms a mix of both defence and offense is key to help predict success.

Similarly, if we are to look at the feature selection to provide info. We can see from the information gain: defensive pressure is very important, but as is passing stats. Defensive pressure may also indicate how hard a team presses versus them sitting back and defending. This isn't an absolute indicator of defensive performance. The kbest feature selection results show a similar story.

Looking at the comparison, it's also clear the defensive model has the lowest average accuracy score amongst all the models. This tells me, if anything, defense is one of the relatively weaker influencers. This is all relative, as the difference between the model are tiny (0.01).

In conclusion, this project aimed to answer the question: "does defense win titles?". Using the performance indicators provided and the "success" indictor of betting odds, we have determined: "sort of". Defense does help win titles, but it is definitely not the only thing that wins titles, nor does it seem to be the most important. A team must focus on all aspects of play and create a balanced presence on the pitch.

A team's focus should also be taken with a grain of salt as, based on the accuracies and stats provided, basing your victory purely off the input stats provided in this project is a bit of a fool's task, as they help influence victory but is not a guarantor. At the end of the day, none of the models created a very good level of accuracy.

Solely focusing on defense will create a team that's slightly less likely to win as compared to teams solely focus on offense, or teams focused on a hybrid approach.

5) Limitations and Recommendations

In this section I will detail the analysis limitations, study implications, ethical considerations, project continuity, and critical insights on this work.

<u>Analysis Limitations/Critical Insights:</u>

There is a massive time and resource constraint on the research as the author had to split his time between this project, family, work, and survival. No offense to the reader, this project did not hold the highest priority over the author's physiological needs as a human. The author is an amateur in the field of data analytics, and as such the path to the results may not be optimal. Given more time, expertise, and physical energy the models could've been improved with further experimentations such as changing the parameters, or a different type of cross validation could be used.

Models and results can be improved upon given the time, resources, and support.

Input data from the project is taken directly from the video game FIFA. As such the stats should be taken with a grain of salt, as there is going to be an amount of bias in the base data itself. This data is curated and created by football "experts" that work at EA (Electronic Arts) for the sake of entertainment and creating a video game to simulate real life. It's not raw data that takes in and describes what has happened in life exactly with no censorship and influence, but rather an interpretation of real life through the perspective of these "experts".

Similar to previous, the output variable are betting odds. These are not 100% rock solid indicators of success. Betting odds are not a guarantee, one can still beat the odds. Like how the input variables are EA's interpretation of real life, the output variable of betting odds is the bookkeepers' interpretation of real life.

Despite the results produced, things should be taken lightly as the input dataset is not built upon a rock solid foundation, but rather excerpt interpretations.

The models created here also only look at home wins. This introduces home team advantage as a possible bias. Including away wins may help with this.

Study Implications:

This study helped confirm what's sort of the popular discourse in football. Which is that one cannot solely rely on purely defense nor offense. To be a successful team that has greater odds of winning, both sides of the field must be balanced.

Ethical Considerations:

Data source is linked and credited:

https://www.kaggle.com/datasets/hugomathien/soccer?datasetId=63&sortBy=voteCount

I do not claim to be the original writer of all the code used in this project. Most of the code is Googled copied and pasted, and jerry rigged together so that it fits the context of my dataset and question.

Project Continuity:

Future suggestions for researchers with more resources:

- Collect more in-depth and more input variables.
- Try other forms of cross validation to change the input training and test sets.
- Build models with different parameters input.
- A more robust measure of "success" can still be developed.

Bibliography

Anderson, C., & Sally, D. (2013). *The Numbers Gae.* Penguin.

Breaking the Lines. (2022, July 28). *What is Juego de Posición?* Retrieved from Breaking the Lines: https://breakingthelines.com/tactical-analysis/what-is-juego-de-posicion/

Brownell, P. (2013, July 31). *Which Stats Are Most Important for Measuring Defenders?* Retrieved from Bleacher Report: https://bleacherreport.com/articles/1722602-which-stats-are-most-important-for-measuring-defenders

Ćwiklinski, B., Gielczyk, A., & Choras, M. (2021). Who Will Score? A Machine Learning Approach to Supporting Football Team Building and Transfers. *Entropy*, 90.

Davis , A., & Suryawanshi, N. (2023, 1 14). *Northwestern Sports Analytics Group*. Retrieved from Does Defense Win Championships?: https://sites.northwestern.edu/nusportsanalytics/2020/12/15/does-defense-win-championships/

FOX Sports. (2023). *FIFA World Cup 2022 Defensive Stats*. Retrieved from Fox Sports: https://www.foxsports.com/soccer/morocco-men-team-summary-stats?category=defensive&sort=t_int&season=2022&sortOrder=desc&groupId=12

MARCA. (2012, 9 12). *Spain beats its possession record for the Del Bosque era*. Retrieved from MARCA: https://www.marca.com/2012/09/12/en/football/national_teams/1347468397.html

Merhej, C., Beal, R., Ramchurn , S., & Matthews, T. (2021). What Happened Next? Using Deep Learning to Value Defensive Actions in Football Event-Data. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 3394-3403.

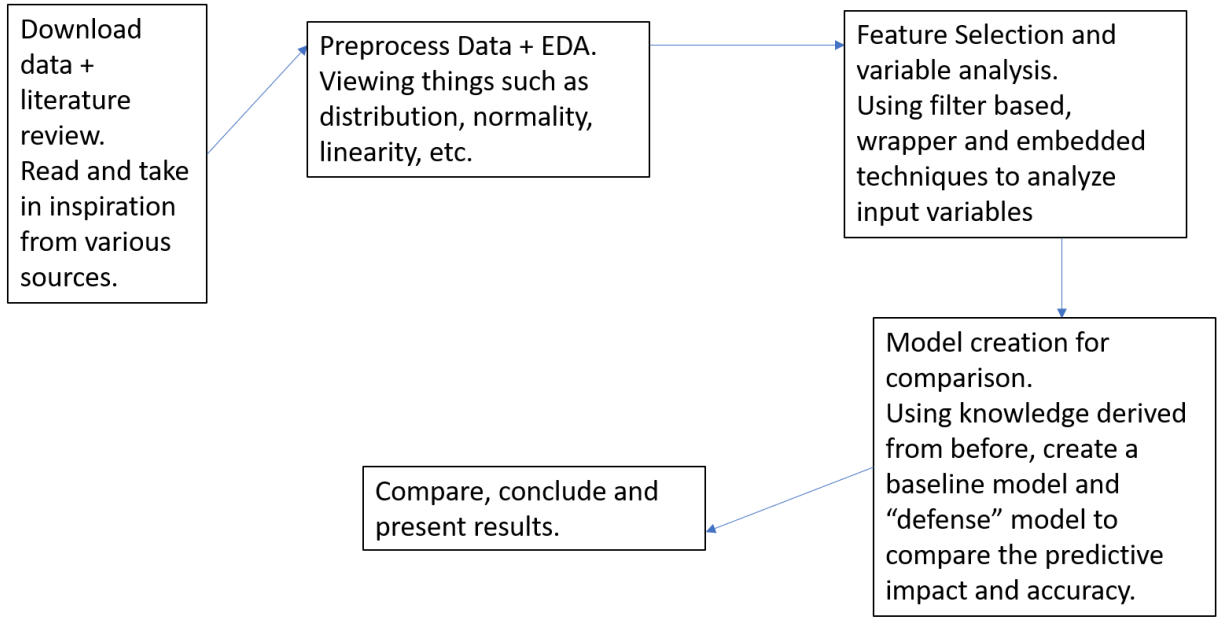
Rossi, A., Pappalardo, L., Cinitia, P., laia , F. M., Fernandez J,, & Medina, D. (2018). Effective injury forecasting in soccer with GPS training data and machine learning. *PloS on*.

Smith, A. (2017, 8 30). *Does attack or defence win titles?* Retrieved from Premier League: https://www.premierleague.com/news/464218

Stöckl, M. S. (2021). Making offensive play predictable-using a graph convolutional network to understand defensive performance in soccer. *In Proceedings of the 15th MIT Sloan Sports Analytics Conference*.

Winterburn, S. (2017, January 7th). *What can we learn from defensive statistics?* Retrieved from Football3665: https://www.football365.com/news/what-can-we-learn-from-defensive-statistics

Appendix A - Proposed Roadmap – Extremely General



Appendix B – Full Decision Tree Sample

Below is a screenshot of the massive decision tree that's generated using the baseline model.

A PNG file as well as a textual version of the full decision tree is included on GitHub for your reference.

As it's too big (text is 70+ pages, image is too wide), it cannot be effectively included on the report.

The purpose of including the tree below is not to show the results of the tree but rather how large and expansive the tree is.

Due to the sheer size of the tree and the scope of the project, I will only be concerning myself with the strongest predictors, aka the variables that would exist on the top of the trees.

Appendix C – Math Behind the Code

The following is one sample calculation to show the math behind the code for the accuracy stats used in the analysis phase of the project.

The following code shows only the calculations for the decision-tree baseline model. The purpose of this is to show the proof of knowledge of math. Hence, only 1 sample calculation is done.

§ SAMPLE CONFUSION MATRIX CALCULATIONS

- For the purpose of showing how the math works behind my code.
- Sample calculation done on confusion matrix for:

DECISION TREE - BASELINE MODEL:

	0 GW	1 LW	2 LL	3 UW
0 GW	5080	1001	95	197
1 LW	1515	2846	766	1371
2 LL	528	1676	1764	1699
3 UW	757	1902	1328	2417

output variables
0 = Guaranteed win = GW
1 = Likely win = LW
2 = Likely loss = LL
3 = Unlikely win = UW

- Given:

$$\text{Precision} = P = \frac{\text{TRUE POS}}{(\text{TRUE POS} + \text{FALSE NEG})} = \frac{TP}{(TP + FP)}$$

$$\text{Recall} = R = \frac{\text{TRUE POS}}{(\text{TRUE POS} + \text{FALSE NEG})} = \frac{TP}{(TP + FN)}$$

① $P_{GW} = \frac{TP}{(TP + FP)} = \frac{5080}{2800 + 5080} = 0.64$

where: $FP = 1515 + 528 + 757 = 2800$, $TP = 5080$

$R_{GW} = \frac{TP}{(TP + FN)} = \frac{5080}{5080 + 1293} = 0.79 \approx 0.80$

where: $FN = 1001 + 95 + 197 = 1293$, $TP = 5080$

② $P_{LW} = \frac{TP}{TP + FP} = \frac{2846}{2846 + 3529} = 0.38$

where: $FP = 1001 + 1676 + 1902 = 3579$, $TP = 2846$

$R_{LW} = \frac{TP}{TP + FN} = \frac{2846}{2846 + 3652} = 0.437 = 0.44$

where: $FN = 1515 + 766 + 1371 = 3652$, $TP = 2846$

③ $P_{LL} = \frac{TP}{TP + FP} = \frac{1764}{1764 + 2189} = 0.446 = 0.45$

where: $FP = 95 + 766 + 1328 = 2189$, $TP = 1764$

$R_{LL} = \frac{TP}{TP + FN} = \frac{1764}{1764 + 3903} = 0.31$

where: $FN = 528 + 1676 + 1699 = 3903$, $TP = 1764$

④ $P_{UW} = \frac{TP}{TP + FP} = \frac{2417}{2417 + 3267} = 0.425 = 0.43$

where: $FP = 197 + 1371 + 1699 = 3267$, $TP = 2417$

$R_{UW} = \frac{TP}{TP + FN} = \frac{2417}{2417 + 3987} = 0.377 = 0.38$

where: $FN = 757 + 1902 + 1328 = 3987$, $TP = 2417$

- Given: $F1\text{-score} = 2 \cdot \left(\frac{P \cdot R}{P + R}\right)$

① $F1_{GW} = 2\left(\frac{P_{GW} \cdot R_{GW}}{P_{GW} + R_{GW}}\right)$

$= 2\left(\frac{0.64 \cdot 0.80}{0.64 + 0.80}\right)$

$= 0.71$

② $F1_{LW} = 2\left(\frac{P_{LW} \cdot R_{LW}}{P_{LW} + R_{LW}}\right)$

$= 2\left(\frac{0.38 \cdot 0.44}{0.38 + 0.44}\right)$

$= 0.407$

$= 0.41$

③ $F1_{LL} = 2\left(\frac{P_{LL} \cdot R_{LL}}{P_{LL} + R_{LL}}\right)$

$= 2\left(\frac{0.45 \cdot 0.31}{0.45 + 0.31}\right)$

$= 0.367$

$= 0.37$

④ $F1_{UW} = 2\left(\frac{P_{UW} \cdot R_{UW}}{P_{UW} + R_{UW}}\right)$

$= 2\left(\frac{0.43 \cdot 0.38}{0.43 + 0.38}\right)$

$= 0.40$

- using the confusion matrix, formulas, and calculations above the following summary is generated for the report.
- Note - support is simply # of observations w/ a TRUE response for each output variable.

	PRECISION	RECALL	F1	SUPPORT
GW	0.64	0.80	0.71	6373
LW	0.38	0.44	0.41	6498
LL	0.45	0.31	0.37	5667
UW	0.43	0.38	0.40	6404

fig. summary table for 'decision tree- baseline model'.