Data warehousing and data mining report

# student details

### Name(s): cathal hughes & russell brady

### Programme: case

### Module Code: ca4010

### Submission Date:

### Module Coordinator: Mark roantree

# Plagiarism declaration

We declare that this material, which We now submit for assessment, is entirely our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of our work. We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. We have read and understood the Assignment Regulations. We have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged, and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me/us or any other person for assessment on this or any other course of study.

We have read and understood the referencing guidelines found at http://www.dcu.ie/info/regulations/plagiarism.shtml, https://www4.dcu.ie/students/az/plagiarism and/or recommended in the assignment guidelines.

Name(s): Russell Brady & Cathal Hughes Date: 24/10/2018

table of Contents

[student details 1](#_Toc529903863)

[Name(s): cathal hughes & russell brady 1](#_Toc529903864)

[Programme: case 1](#_Toc529903865)

[Module Code: ca4010 1](#_Toc529903866)

[Submission Date: 1](#_Toc529903867)

[Module Coordinator: Mark roantree 1](#_Toc529903868)

[Plagiarism declaration 1](#_Toc529903869)

[Introduction 2](#_Toc529903870)

[Idea and dataset description 3](#_Toc529903871)

[Idea 3](#_Toc529903872)

[Dataset Description 3](#_Toc529903873)

[Premier League Games Dataset 3](#_Toc529903874)

[PL Standings Dataset 4](#_Toc529903875)

[Generated Dataset 4](#_Toc529903876)

[Data Preparation 5](#_Toc529903877)

[Attribute Analysis 5](#_Toc529903878)

[Data Transformation 8](#_Toc529903879)

[Attribute Construction & Aggrergation 8](#_Toc529903880)

[Data Integartion 11](#_Toc529903881)

[Premier League Standings Dataset 11](#_Toc529903882)

[Discrepancy Detection 11](#_Toc529903883)

[Missing Values 12](#_Toc529903884)

[Final Dataset 13](#_Toc529903885)

[Algorithm description 14](#_Toc529903886)

[The Implementation 15](#_Toc529903887)

[Results and Analysis 17](#_Toc529903888)

# Introduction

The following report provides a detailed explanation of our idea and dataset, how we prepared our data, the algorithm we used to make our prediction and our subsequent results and findings. Also included is our insights on how we felt the process went and our general thoughts on the results and predictions with which we obtained.

# Idea and dataset description

Soccer is the most popular sport in the world and is played by 250 million players in over 200 countries and dependencies. The premier league is arguably the most popular league in the world with huge financial strength, TV coverage and entertaining matches. There are also vast amounts of data available in datasets online relating to the premier league with everything from match results to the number of throw-ins in a game. Because of this we decided it would be an appropriate area to apply data mining techniques and make predictions.

## Idea

Our idea was to take the statistics and results from the last number of years in the premier league, clean the data to obtain only the attributes we wanted, pass this data through an appropriate algorithm and generate a prediction for the result of a given game based on this. Our hope was to predict whether a game would end in a home win, away win or a draw.

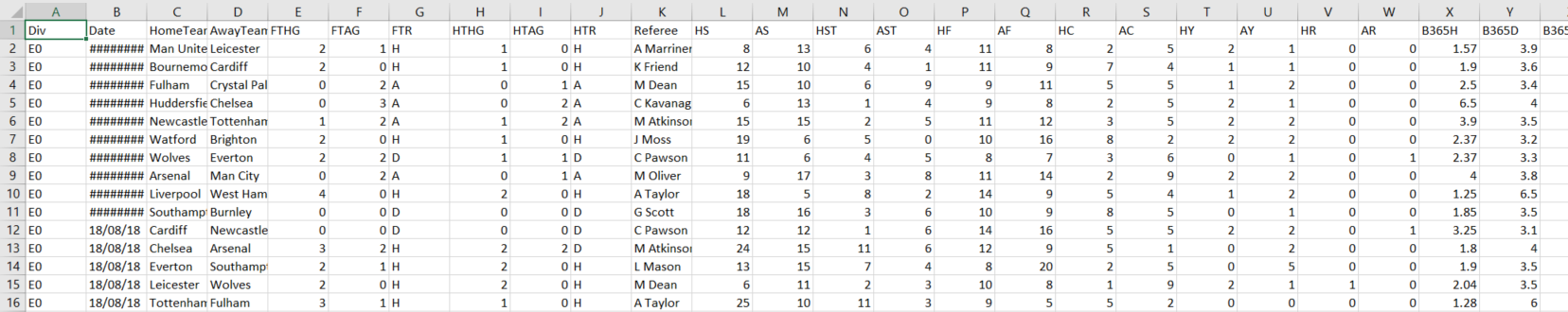
The reason we decided to take on this idea is that there is a real-world application. Betting is a huge part of all sports, soccer being no different, with up to one billion euros being placed on each premier league games globally according to the Tribune. Our idea was that if we could generate predictions with a reasonable accuracy there would be a possible monetisation opportunity. With a reasonable prediction accuracy for example, you could have a bot place smart bets across a host of games to try and make money. You would only place bets across games where the odds, along with our predictions would indicate you are going to at least make your money back. Obviously, this is all theoretical but we were curious to find out the possibilities and that is why we went with the idea.

## Dataset Description

Although there was plenty of data available, it became apparent that we weren’t going to be able to obtain a perfectly clean standardised dataset. This meant we were going to have to generate our own dataset using cleaning techniques on existing data for premier league games and generate our own data for premier league standings for the relevant years.

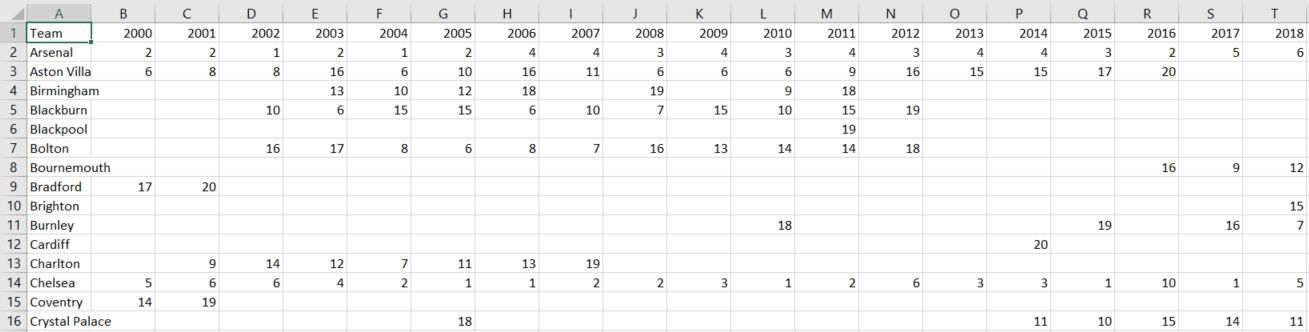
### Premier League Games Dataset

These datasets were obtained from www.football-data.co.uk/data.php. Each dataset was split up into its respective season and contained stats about all the matches in that given season. Each dataset contained a different number of attributes some contained 28 while others contained 65, some relevant and some not. Also, there was 380 rows per dataset since there are 20 teams, each playing each other twice during the course of the season. We collected the datasets from the 2000/01 season to the 2017/18 season. This meant we had 6840 rows of data to work with. The picture below is a snippet of some of the pre-cleaned data.



### PL Standings Dataset

We also used a dataset pertaining to Premier League standings for the relevant years. This is a dataset we generated ourselves. It was arranged alphabetically by team and contained the teams finishing place for the last 18 years in the Premier League. The dataset included the 46 teams which were involved over the course of the last 18 years. This dataset also contained multiple empty values for newly promoted teams and teams that haven’t been in the Premier League in years e.g. Coventry. This dataset would need to be incorporated with the above datasets and all empty values would need to be handled appropriately. The picture below is a snippet of our PL Standings Dataset.



### Generated Dataset

After concatenating all the datasets and incorporating the Premier League Standings dataset we then had to perform some major cleaning on the dataset. This will be discussed in greater detail further on in the report.

The quantity and differing numbers of attributes in the PL Games Dataset was difficult to deal with and was something that would not be feasible to work with. Some of these attributes related to in-games stats and bookies odds. We had no use for in-game stats as it would be pointless to make a prediction on a game in the middle of one. Although, the bookies odds could have proved useful and could have improved our prediction accuracy, we did not have the bookies odds for much of the dataset and thus we could not use them to make our predictions. As such we decided that we wanted to make our prediction solely based on the two teams and their form. The results would not be swayed by any third party and all predictions could be made using some basic stats of the two teams.

There are some attributes which weren’t relevant on their own but proved useful when used in conjunction with others meaning we had to generate some of our own attributes which will be explained in greater detail below. After cleaning, we were left with a dataset which contained 6840 rows and 12 attributes.

These attributes were the following - Home Team Points, Away Team Points, Difference in League Position, 6 columns containing team form over the last three games for both teams, home and away team goal difference as well as difference in league points. The target label for the dataset was Full Time Result which is one of these three values - Home Win (H), Away Win (A) or Draw (D). This was a clear classification problem.

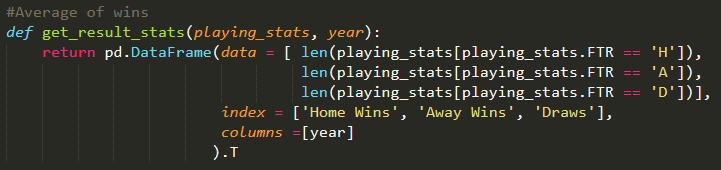
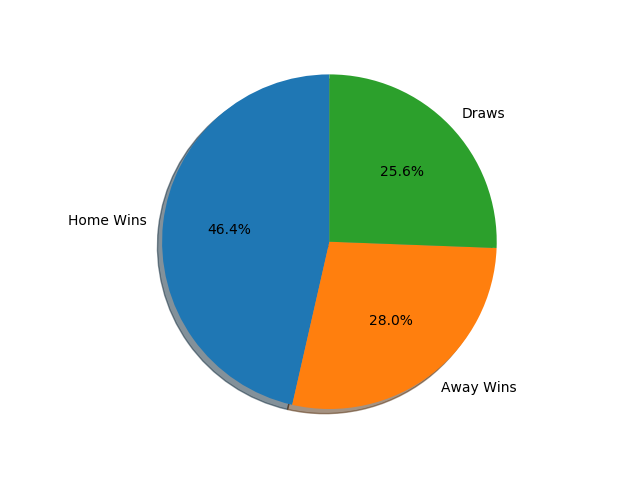
Generating our final dataset to be passed into our algorithm was a challenge in terms of deciding which attributes to use, and then extracting these attributes. Our data contained categorical labels and only some of the data was standardised, meaning how we cleaned the data was very important. Once we generated the final dataset we then split it up into a training and testing set. Our training set contained 17 years of data from 00/01 to 16/17 and our testing set contained the most recent full season, the 17/18 season. Although this was not an optimal split it allowed us to make predictions for one whole season.

# Data Preparation

All the data was contained in CSV files and all the data cleaning was performed using python. Tools such as Pandas and ScikitLearn were used for manipulating and concatenating the data. In order to get to our final dataset we had to do some major cleaning of the data. The first step of this was identifying the attributes we wanted to use in order to predict the target label. The attributes we chose needed to be achievable to generate using the data we had at our disposal. We separated the cleaning up into three different scripts namely - cleaning\_data.py, final\_dataset\_cleaning.py and data\_preperation.py.

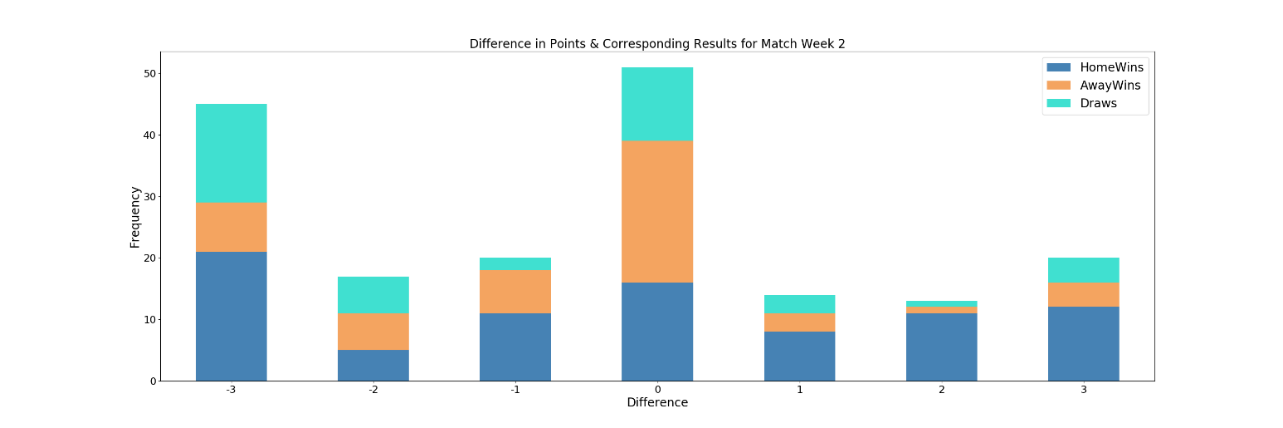
## Attribute Analysis

In order to evaluate which attributes could contribute to our prediction we needed to perform attribute analysis. The first attribute we analysed was the aggregate win percentage across all the 18 years. We overlooked how big a part home advantage played. We quickly learned that there should be a bias towards the home team due to the fact 46.4% of the games in our dataset attributed to Home Wins. 28 % of the games were Away Wins whilst 25.6% of the games ended up as a Draw. This is the cod used to generate these stats:

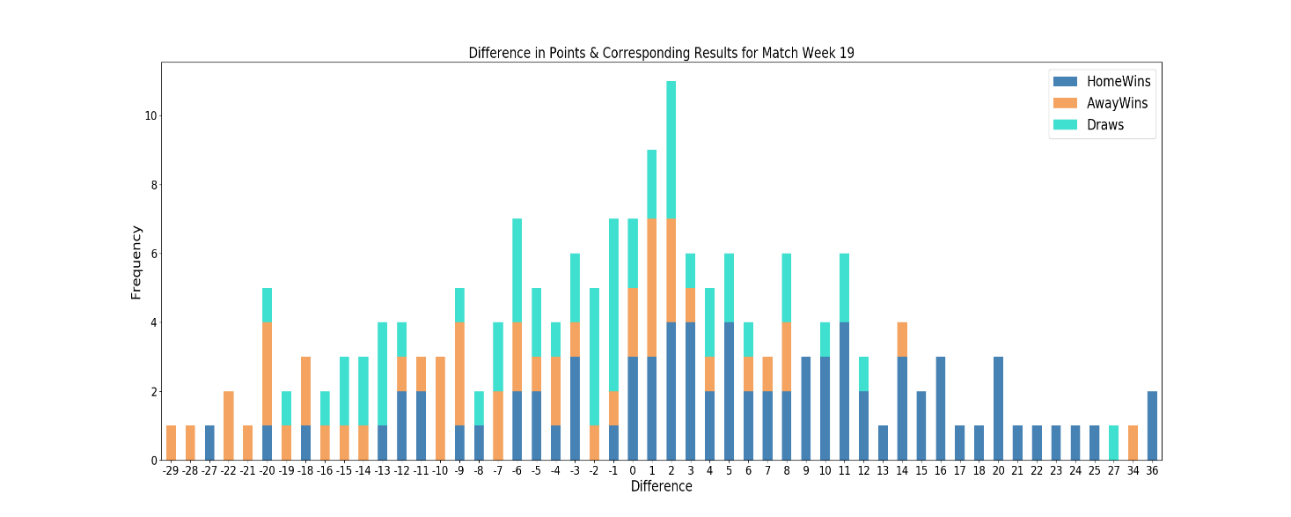
 

Another potential attribute we could generate, and use was points difference between teams. Early on in the season this stat has no bearing as the League leaders could easily have the same points as a team in tenth position. As the weeks and games progress though, it seems like it has a huge bearing on the final result. The greater the difference in points between the teams the more likely the team with the greater amount of points would win.

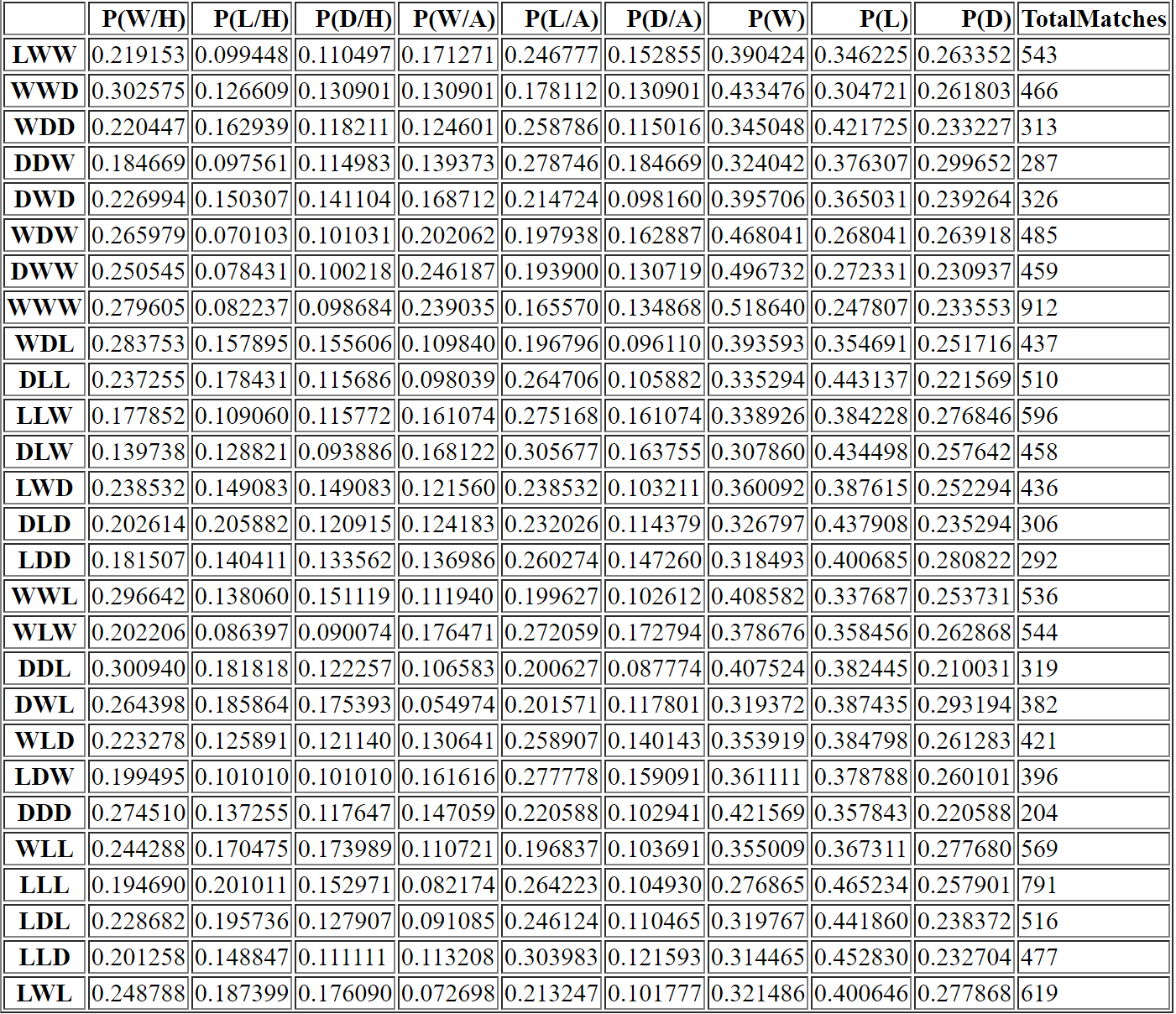
This graph shows how points difference doesn’t really have an impact in the early weeks of the season:



This graph shows that as we reach the midpoint in the season, after more games have been played, that the gap between teams has a greater effect on the result. Its also clear that when there is little or no difference in points between the teams, there is a more even match up. A home win is still the most likely result but the chances for an Away win or Draw are increased. This is quite evident at a difference of +2 in the chart below.



Form is a major factor heading into a game of football and we knew this is another attribute we could generate and potentially use. We analysed a team’s form to see what effect it could have on a result . We used the results of the last three games for both teams and created probabilities of a win, loss or draw overall and then for whether the team was home and away. This meant we had every permutation of win, lose, draw for a team’s last 3 games and we had 9 probabilities for every permutation. The probability chart can be seen below.

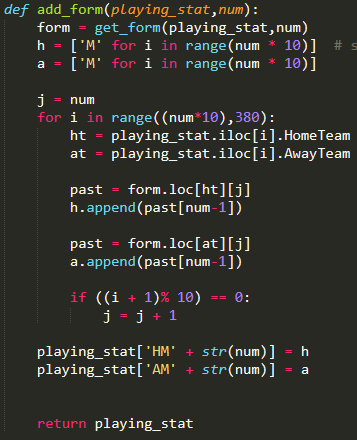
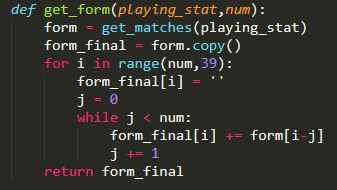


# Data Transformation

## Attribute Construction & Aggrergation

Based on the analysis we performed we generated the following attributes: Home Team Points, Away Team Points, 7 attributes pertaining to the results of the last 3 games for both teams, Home Team Goal Difference and Away Team Goal Difference.

Attribute Construction is where new attributes are constructed and added from the given set of attributes to help the mining process. We used this to generate the form for both teams across their last three games. This attribute was constructed from the date and full-time result attributes already in the dataset. Here is some of the code used to generate the form.



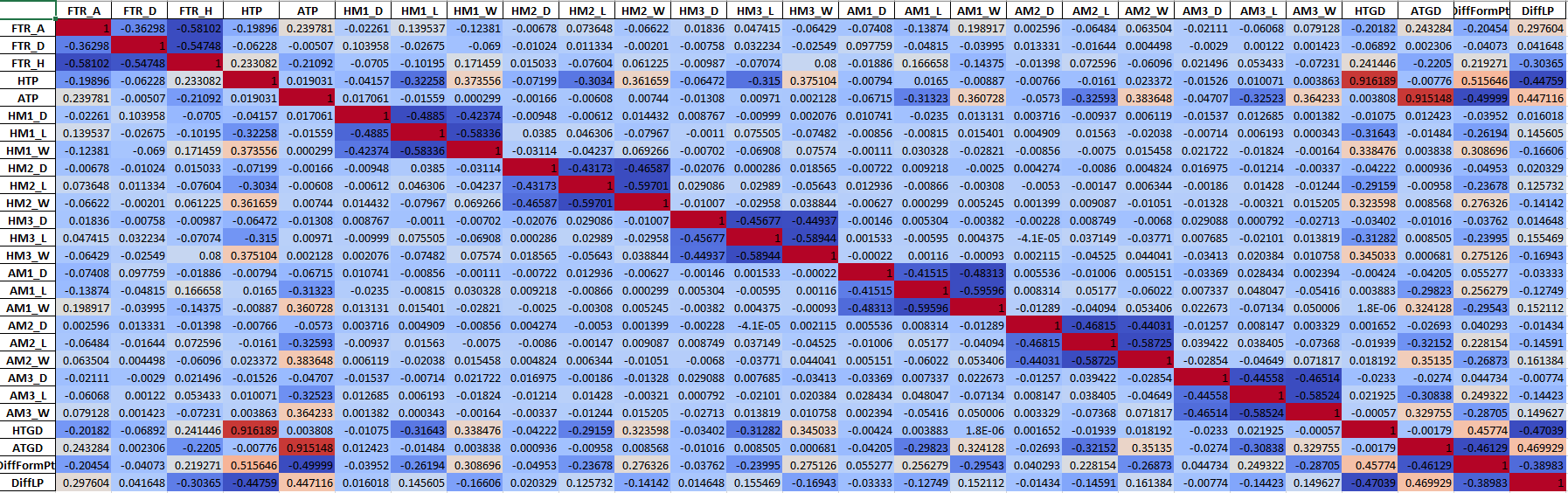
Aggregation is where summary or aggregation operations are applied to the data. We used aggregation to create Home Team Points and Away Team Points. This was an aggregation of all the results for a team in that season. Similarly, we created Goal Difference for both teams by getting the goals scored and conceded by a team at that point in a season. Goals scored by both teams were two attributes in our original dataset and so aggregation made it possible to create these new attributes.

Here is how we aggregated goals scored and conceded in a game to create goal difference for both teams.

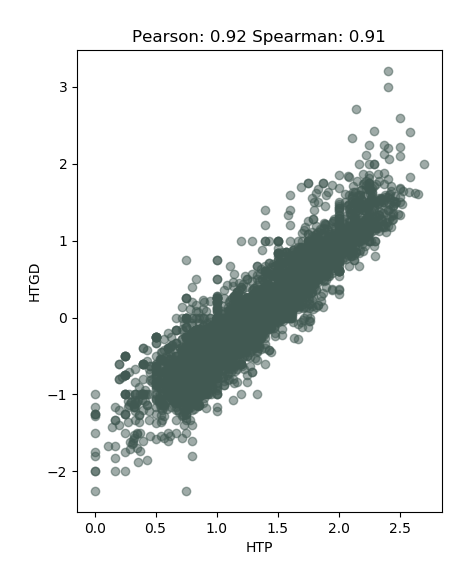
A screenshot of text

Description generated with very high confidence

After creating all these new attributes, we decided to perform a correlation analysis to try catch any redundancies in our new attributes. Given two attributes we knew such analysis could measure how strongly one attribute implies the other. We calculated the correlation coefficient between every attribute and analysed this data to see if we could spot any potential redundancies. Here was the output we had to work with:



Straight away we could see there is a strong positive correlation between home team points and home team goal difference as well as a strong correlation between away team points and away team goal difference. We analysed these relationships further and found that there was a Pearson score of 0.92 between Home Team Points and Home Team Goal Difference. The correlation can be seen below:



This means we could potentially remove either of these attributes from our newly created attributes. Perhaps we could make predictions using both attributes in the dataset and then make two more predictions where we remove just Home Team Points/Away Team Points and in the other prediction we remove Home Team Goals/Away Team Goals.

# Data Integartion

Data integration involves combining data from multiple sources into one coherent source. We used data integration in coming up with our own premier league standings and incorporating it into our final dataset of attributes.

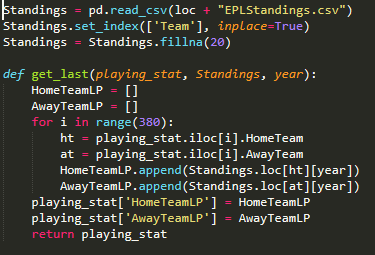
## Premier League Standings Dataset

We created this dataset pertaining to premier league standings for the relevant years which were covered in the Premier League Games dataset. This contained standings for 46 teams which participated in the competition in that time.

The reason we chose to create this dataset was that we saw the difference in league position from the previous year as a possible attribute which could be used to distinguish the stronger team, and in turn who would be most likely to win the match.

We felt this attribute would work particularly well with teams that are forerunners in the premier league and those who are near the bottom of the table, or just newly promoted. For teams who were in and around the same point, it would indicate a fair match up and wouldn’t have as much of an impact in the prediction.

To ensure we weren’t introducing a redundancy we performed a correlation analysis. The analysis showed that there was no clear redundancy between any of the attributes, so we were happy to integrate this into our dataset. The code below is what we used to integrate the dataset into our final dataset:

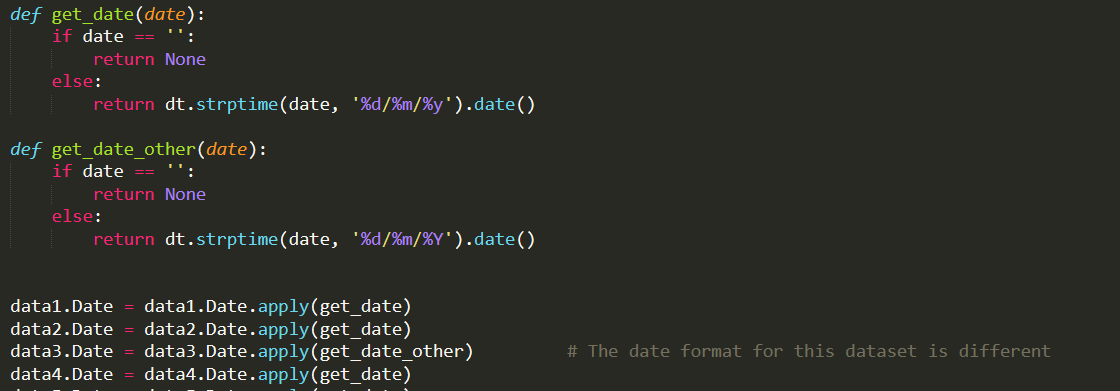


This dataset contained missing values due to teams being relegated and not being in the premier league for years. This is explained in further detail later in the report.

## Discrepancy Detection

The first step when cleaning our data was to find discrepancies between our 18 Premier League Games datasets. This involved finding inconsistency between the datasets and dealing with them appropriately. The first data discrepancy we came across was the varying number of columns between datasets. As mentioned above, some of the datasets contained 28 columns while others contain up to 65. For example, our 2001/02 season dataset contained 28 whilst the 2017/18 dataset contained 65. The extra columns for the 2017/18 season pertained to bookies odds. As these extra columns across every dataset, we removed them.

When we initially started working with datasets we realised we would need to use the date of games to generate form for each team. The form was related to the teams’ last three results. Initially we tried to get the date for each match in each dataset using a simple “get\_date” method. After trying to apply this function to every dataset we realised that an issue was occurring with the 2002/03 dataset and the “get\_date” method was not parsing the date correctly. The date in this dataset was formatted differently and needed to be handled using a different function.



The issue was that in the 2002/03 the year was represented as a year with century as a decimal number, e.g. 2002 or 2003. In all the other datasets the year was represented as so: 00, 01, 02 etc.

## Missing Values

As we generated the Premier League Standings dataset ourselves, we knew it contained missing values. As mentioned above, it contains teams that have not been in the Premier League in years, newly promoted teams and teams that have been recently relegated. As they are only 20 teams in the Premier League in a given year and we had 46 teams in the dataset, there was plenty of missing values. To overcome this issue, we decided we would use a global constant to fill in the missing values. This would mean filling in all the missing values with the same value. This method is biased to the data as the constant used is not the correct value, but we felt it was the most suitable. Any entry in the dataset was in the range from 1-20 which was where the team had finished in the league for that year. For the missing values we used the constant 20. This was the lowest ranking for a team and allowed us to integrate the dataset with the Premier League games dataset.

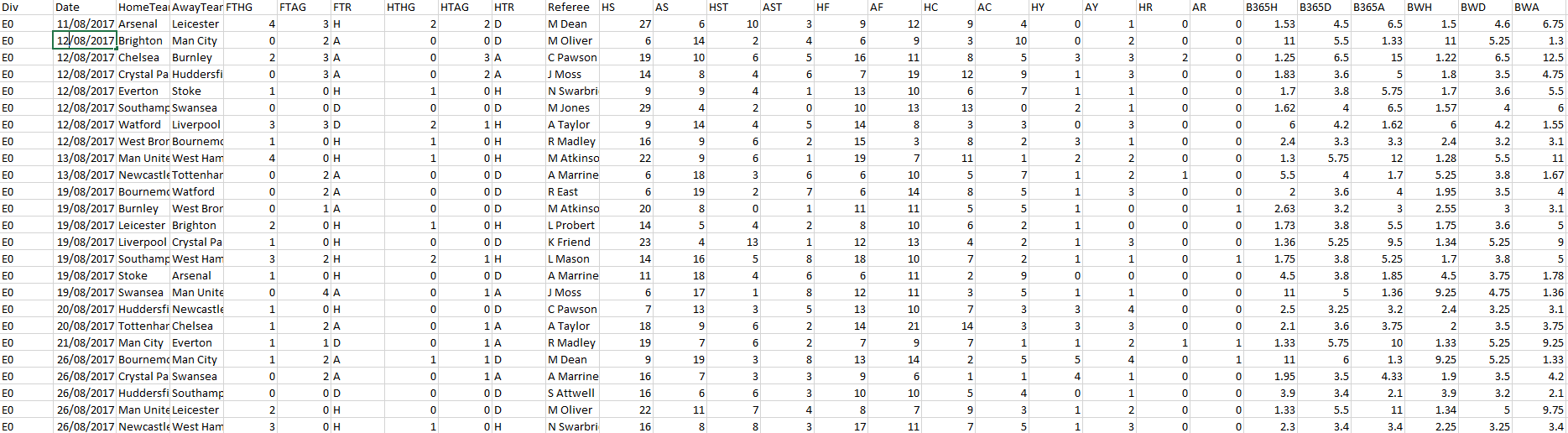
Another way we could have filled in the missing values was Generalisation. The teams could have been ranked in blocks e.g. 1-5, 6-10, 11-15, 16-20. This would

As we started generating our own attributes for the dataset it was clear that team form was going to be one of attributes we would use to make a prediction. We decided to use the results of the teams last 3 games as their form. We had to remove each team’s first three games in the dataset as it would not be possible to generate form for a team in its first three games. The method used to handle these missing values was just to ignore the first three match-weeks for each team in each dataset.

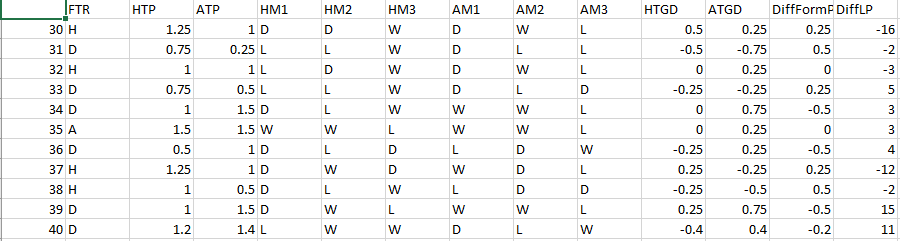
## Final Dataset

Based on our analysis of the data and subsequently our data transformation and integration, we generated key attributes which will be central to our prediction algorithm. We started with 18 datasets from the years 00/01 to 17/18, which had varying numbers of attributes (most of which were irrelevant to a prediction of the winner of a game due to the fact they were in-game stats). Using appropriate data integration and transformation techniques explained above we managed to create one single dataset with a set of relevant attributes. Here is a comparison of our dataset before and after our analysis and these techniques were applied:

**Dataset before:**



**Dataset after:**



We now have a set of attributes that are suitable for a classification algorithm.

# Algorithm description

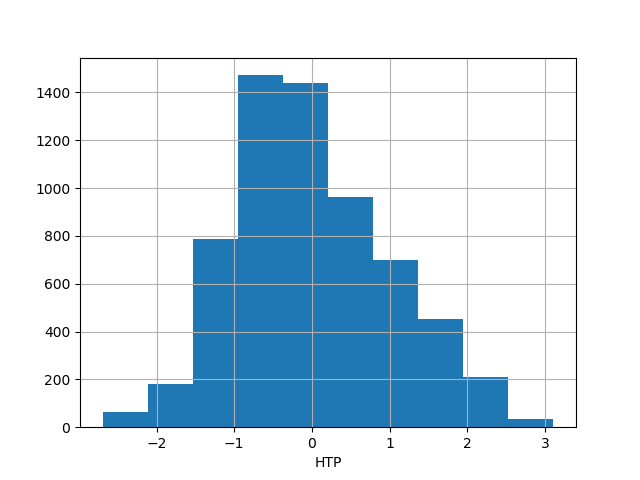
To choose the most appropriate algorithm we first needed to make sure we understood our data as the type and kind of data plays a key role in deciding which algorithm to use.

Having performed attribute analysis and subsequent data transformation and integration we were left with our final dataset containing our input and target output. In choosing our algorithm we had to categorise our problem and see what type of data we were dealing with.

In categorising our input, we knew our data was labelled and so pertained to supervised learning. We also knew we were dealing with a classification problem as the output of the model is a class. Both of these are factors in choosing the correct algorithm. There were a number of algorithms we could have chosen from, the main ones being Naive Bayes, KNN and a Decision Tree.

From early on though, it seemed as though a Decision Tree or Naïve Bayes would be the most suitable. One of the major advantages of Naïve Bayes is that it does not care if there is relationship or a correlation between the attributes. This suited our dataset as there was a strong positive correlation between Goal Difference and Points a team had. This mean there was a redundancy in the data and we could potentially lose two attributes trying to remove this redundancy. Naïve Bayes would not have been swayed by this strong positive correlation though and thus it became a potential candidate for our chosen algorithm.

As our dataset was quite small (6840 rows) and 46% of these rows were labelled as a ‘H’ indicating a Home Win we felt it would be best to use a classifier with. As our attributes are normally distributed we felt that Gaussian Naïve Bayes was the most suitable classifier to use for this data. These images below show the normal distribution of some of our attributes.

A close up of a logo

Description generated with high confidenceA picture containing text

Description generated with high confidenceA close up of a logo

Description generated with high confidence

We had to take into consideration the bias-variance trade-off. Upon evaluating our final dataset, we knew couldn’t afford high variance on such a small amount of data due to the possibility of overfitting.

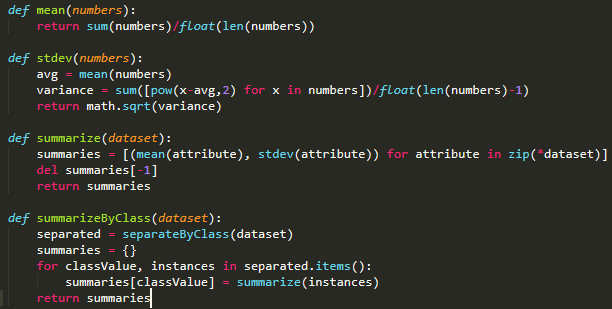
<https://pdfs.semanticscholar.org/2908/0eeec394d5958a2baf9d779491540130b6e7.pdf>

## The Implementation

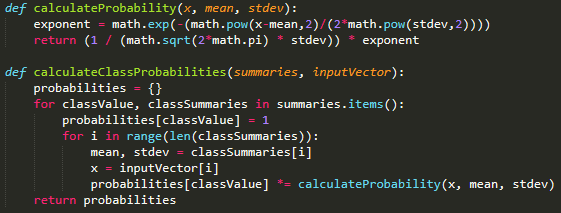
The initial step when running the algorithm on our dataset was to load the dataset and then split it into a training and testing set. Once the dataset is loaded and all the string values have been converted to floats, it is passed into split dataset with a splitRatio, in our case we used a splitRatio of 0.85. This means 85% of the data will be used for training whilst the other 15% will be used for testing.



The data is then summarised. This summarised data is what makes up the Naïve Bayes model and consists of the standard deviation and the mean of each attribute by class value. This means we first have to separate the data by class and then calculate the standard deviation and mean of the attributes in that class.



Now that we have a summary for the dataset we can now make predictions using the model. As we are using Gaussian Naïve Bayes the first step in making the prediction is to calculate the Gaussian Probability Density Function. This involves calculating the probability of an attribute belonging to a class, applying this to all attributes in an instance, and then coming up with a probability that the entire instance belongs to the class.



Using these generated probabilities, we can return the highest probability for an instance belonging to a class. I.e. is the highest probability for the instance associated with “H (Home Win)”, “A (Away Win)” or “D (Draw)”. This is our way of making a prediction. Using these predictions on a test set we can check the accuracy of our trained model to see how it performed on unseen data.

A picture containing text

Description generated with high confidence

# Results and Analysis

Detailed in the section above, is how we implemented our own Gaussian Naïve Bayes Classifier. Training this classifier on 80% of our data and testing on the other 20% yielded results of approximately 53%.

This accuracy is quite low and not what we were expecting after performing the relevant analysis and transformation to produce our attributes. We decided to check the prediction accuracy with some of the built in classifiers used in python using the SkLearn classifiers. How easily these classifiers are instantiated can be seen below, along with the accuracy each of the classifiers produced.

A close up of text on a black background

Description generated with high confidence

As you can see Naïve Bayes was the top performer again for the data, albeit at a similar accuracy as our own NB classifier. Although there was comfort to be gained from the fact that we had picked the right classifier, it still begs the question, why the accuracy is consistently low across all three.

We decided to further investigate this, by training a neural network on our data. This again the yielded very similar accuracy to our own Naïve Bayes classifier. Underwhelming, as we hoped some classifier would provide us an accuracy of ~60%.

Perhaps, there is an underlying issue in our data that another workshop would have highlighted. Although an accuracy of 54% should not be underestimated as it is an increase from a guess of 33% for home win/away win/draw and greater than the 46% for guessing home win for every game, we still aren’t satisfied with the results that our final classifier produced. For someone following the game, intuition would provide them with similar results for picking a home win, away win or a draw.

Without access to more stats related to the game, to improve our results, incorporating something like sentiment analysis could really improve our prediction. Also, incorporating bookies odds could improve our prediction, as this would tell us the favourite for matches, this would especially help for games between teams that are in the middle of the table.

From our analysis and the use of our stacked histograms, it is clear that games at the start of the season and games amongst sides in close in position in the table are harder to predict due to points, goals difference and league position all being very similar. The predictions made for these games are likely to be decreasing the accuracy of our classifier.