Capstone Project Interim Report
Patrick McGuinness

Project objective:
Creation of a form-based predictive model for football match results

Overview
The goal of this project is to create a high-performing model based on previous results of football matches. As an ultimate commercial objective, the idea is outperform the betting markets, to this end the client for this project is myself and also Paul Lamford of Phoenix Probability, London who has expressed an interest in my work. It is not anticipated that this project in itself will result in such a model, however it is hoped that it will provide respectable results and insight, and create avenues for future research, or be able to be further extended or incorporated into an existing model for commercial use.

Question: how can simple measures of form, available in previous results or league tables, best be utilised as predictors? Does it matter how a team wins, or just that they win? To what extent can the predictive variables be simplified by combination?

Data source:
The data source used is www.football-data.co.uk

Initial work was to concentrate on one league, the English Premiership (the data source features many leagues from many nationalities, and a possibility for development is to expand the data).

Phase 1:
Initial Data Wrangling and Cleaning
All premiership matches were downloaded, since the start of the premiership in 1998-1999 season. These were well-compiled the only awkward feature was that they were in distinct files for each season, and each file had the same name. Preliminary work involved splicing together the files, selecting the columns to be preserved, and creating a new variable, Season, to distinguish the differing seasons.

Blank rows were removed and the result variable FTR (a factor) was reworked so it was stored in the format (1 - "H", 2 - "D", 3 - "A") for Home Win, Draw, and Away Win respectively rather than in the inverse thanks to default alphabetical. This proved necessary for later ease of analysis.

The resulting dataframe premiership1 and file premiership.csv features the following variables:

premiership1 dataframe
structure

Season	integer. 1 for 1998/99, 2 for 1999/00, … 18 for 2015/16
Div	factor. "E0" for premiership. This variable was kept for future expansion.
Date	factor. The date of the match. Kept for reference and future development.
HomeTeam	factor. The Home Team
AwayTeam	factor. The Away Team
FTHG	integer. "Full Time Home Goals"
FTAG	integer. "Full Time Away Goals"
FTR	factor. "Full Time Result" 3 levels "H", "D", "A"

I initially attempted to separate the seasons automatically by using the date field, however this proved troublesome due to the split of the seasons over two calendar years. After some consideration I realised it was little trouble to manually attach a season integer for each file so this proved to be the simple solution.

Phase 2
Additional variables
Creation of table snapshot for each match

To comprise a base for analysis it was decided to recreate the league table information for both the Home and Away teams for each match at the time the match was played.

To avoid confusion the home team is known as Team1, and the away team as Team2, in the variable descriptors.

The additional variables:

Team1P	Home Team matches played in current season
Team1HomeW	... matches won in current season at home
Team1HomeD	... matches drawn at home
Team1HomeL	... matches lost at home
Team1HomeGF	... goals for at home
Team1HomeGA	... goals against at home
Team1AwayW	... away wins
Team1AwayD	... away draws
Team1AwayL	... away losses
Team1GD	… overall goal difference
Team1Pts	… points

also respective variables (Team2P, …) for the away team.

Creation of this proved somewhat awkward and perhaps resulted in less than elegant code in places, however the job was accomplished.

A 'for' loop was used to take each season in turn, i.e. splitting the data back into individual season, and the new variables were added thus for each season in turn, and the resulting dataframes combined together into one new dataframe object, premiership2.

The home wins, draws, losses, goals for and goals against for the home team (Team1) were compiled using dplyr summary calls (cumsum). This also worked for the away team's away record.

However this approach could not work for the home team's away record or the away team's home record. To create the more difficult half of the form pairs of nested FOR loops were used:

the first pair of FOR loops considers the home team. The first loop goes through the matches, starting at the second. The second loop works backwards and attempts to find the same team in the AWAY position. The away form is then taken from the earlier match, and transplanted forward in time to the new match (combined with the match result from the earlier match).

Similarly, the second pair of FOR loops accomplished the same but considers the AWAY team and locates the last match where the away team played, and obtains the form from there.

The resulting dataframe is then added to premiership2.

Phase 3

Creation of final tables
This was designed to be useful as a validation of calculations and also as a possible source for past season variable analysis

final_tables_home and final_tables_away were created using a join call from the dplyr package with summary and mutate functions and then merged to create final_tables.

Variables created for final_tables include:

HomePlayed
HomeW
HomeD
HomeL
HomeGF
HomeGA
AwayPlayed
AwayW
AwayD
AwayL
AwayGF
AwayGA
GF
GA
Pts

these variables echo a format that can be found commonly in sports publications.

more variables and
Creation of features

These are simply calculated using mutate() calls from the existing variables.

Team1HomeP	Home Team home matches played
Team1AwayP	… away matches played
Team2HomeP	Away Team home matches played
Team2AwayP	… home matches played

these variables then provide the basis for:

T1HWr	Home Team Home win ratio
T1HDr	Home Team Home draw ratio

T1HLr	Home Team Home loss ratio
T1AWr	Home Team Away win ratio
T1ADr	Home Team Away draw ratio
T1ALr	Home Team Away loss ratio
T2HWr	Away Team Home win ratio
T2HDr	Away Team Home draw ratio
T2HLr	Away Team Home loss ratio
T2AWr	Away Team Away win ratio
T2ADr	Away Team Away draw ratio
T2ALr	Away Team Away loss ratio

also created are the dummy variables for the result factor FTR

ResH	1 if FTR = “H”, 0 otherwise
ResD	1 if FTR = “D”, 0 otherwise
ResL	1 if FTR = “L”, 0 otherwise

these contribute to dataframe - premiership2

Initial Explorations

The first few models are binary logistic regressions with the response variable ResH (i.e. if the match is a home win) based on many of these features.

An issue is that early on in the season there is volatility in many of the features so filters were taken at various points in the season. The logistic regression coefficients seemed close at 15 and 20 matches, so a filter of 15 matches played was used for preliminary work.

New Feature

T1GDr	Home Team goal difference per game (current season)
T2GDr	Away Team goal difference per game (current season)

In model8 Goal Difference per game (T1GDr, T2GDr) was introduced to the other variables (the win/draw/loss ratios) and appeared to surpass the others making them redundant.

Model9 & model10 removes the win/draw/loss ratios from the formula just relying on T1GDr and T2GDr as predictor variables. Model10 achieved the lowest AIC thus far – the conclusion is that the goal difference surpasses the win/draw/loss ratios as a predictor and the latter variables are redundant.

Model11 new features was created:

T1Glr the log ratio of goals scored / goals conceded for home team
T2Glr …. away team

looking at the summary and coefficients my conclusion is that the ratio of goals scored / goals conceded is a junk measure compared to the straight goal difference ratio per game and thinking about it this made intuitive sense, it would be a poor football team that won 1:0 half the time compared to one that wins 2:1 on a regular basis: it is the goal difference between the teams that counts to the margin of victory after all.

Model12 trials the combined feature Gdrdiff, which for simplicity is that amalgamation
gdrdiff = T1GDr – T2Gdr

This resulted in the lowest AIC yet

This provides a very simple basis as a predictor variable!

Chasing previous seasons

So given that the vast mass of features appears to be led by one standalone predictor, GDrdiff, the difference between the goal difference per game of the home and away team, how far back is this measure useful?

To this end I added in a similar variables to GDrdiff for the previous seasons,

GDrdiffLS
GDrdiffLS2

This was accomplished using two right_join calls from dplyr to lookup the previous season's figures from final_tables.

Inclusion of GDrdiffLS variable provided significant improvement to model12 and model13. The inclusion of GDrdiffLS2 also provided improvement, but less significant and with a much smaller coefficient. My conclusion is that the data from the season before last is also useful however data before that is likely past its 'analyse-by' date for form purposes, although I have not explicitly checked this.

Imputation of missing values

There were some sources of NA values due to the introduction of previous seasons. A mixture of methods was used to handle these.

1) Previous season's data not available for the first season.
This was handled by omitting the data for the first season.

2) Data from two seasons' ago not available for the second season
As it seems less significant I simply copied the figures from the last season.
i.e. GDrdiffLS2 <- GDrdiffLS

3) In the first match of the season, GDrdiff is not defined
0 was imputed

4) Promoted teams do not have goal difference figures for the previous seasons
To keep the data the mean goal difference for the relegated teams over all the seasons was imputed. This was found to work more smoothly than imputing the mean goal difference for the relegated teams for the particular season in question.