WHO'S GOING DOWN?

Predicting Relegation from the Premier League

By Raja Basu Roy, September 2017

AIM

- To predict who will get relegated from the Premier League using information contained in the current league table

- For example, games won, lost and drawn, goals for and against, number of points, position, current form …

WHAT RELEGATION FEELS LIKE

SO, AFTER SIX WEEKS …

… doing the Capstone, how do I feel?

MILD DEJECTION

RE-EVALUATION

WHO'S GOING DOWN?

Is it possible to predict relegation from the Premier League using the league table?

Yes and no

TWO SECTIONS, TWO STEPS

- Section One
  - Preparation of feature set
  - Analysis of results
- Section Two
  - Preparation of more features
  - More analysis of results

RAW DATA

- Historic league tables throughout the season are not recorded. However, historic match results are and league tables can be derived from them

- Match results obtained from football-data.co.uk

  - Each match has, date, name of home team, name of away team, home goals and away goals

FEATURE ENGINEERING

- Created a function to generate the final league table of each season in order to identify teams that were relegated - the target
- Created another function to produce a league table after N games. N varies between 1 and 38. This is an artificial table as it means disregarding results for teams that have played more than N games while waiting for teams that have played less Premier League games, e.g. due to European and Cup commitments, to catch up.

INITIAL FEATURES

- Decided to have 38 different models that varied with N rather than one that took N as a feature
- Dataset ended up as a dictionary of 38 different dataframes with each dataframe containing 20*22=440 rows where 20 is the number of teams in the league and 22 is the number of seasons after 1995/96 for which there were 20 teams

INITIAL FEATURES …

- Row contains: Games won, drawn and lost, goals for, and against, goal difference and number of points. Treating these as continuous variables

- Hoping to be able to predict relegation from these features

- Saved features to a JSON file. Required a bespoke encoder for a dictionary of dataframes

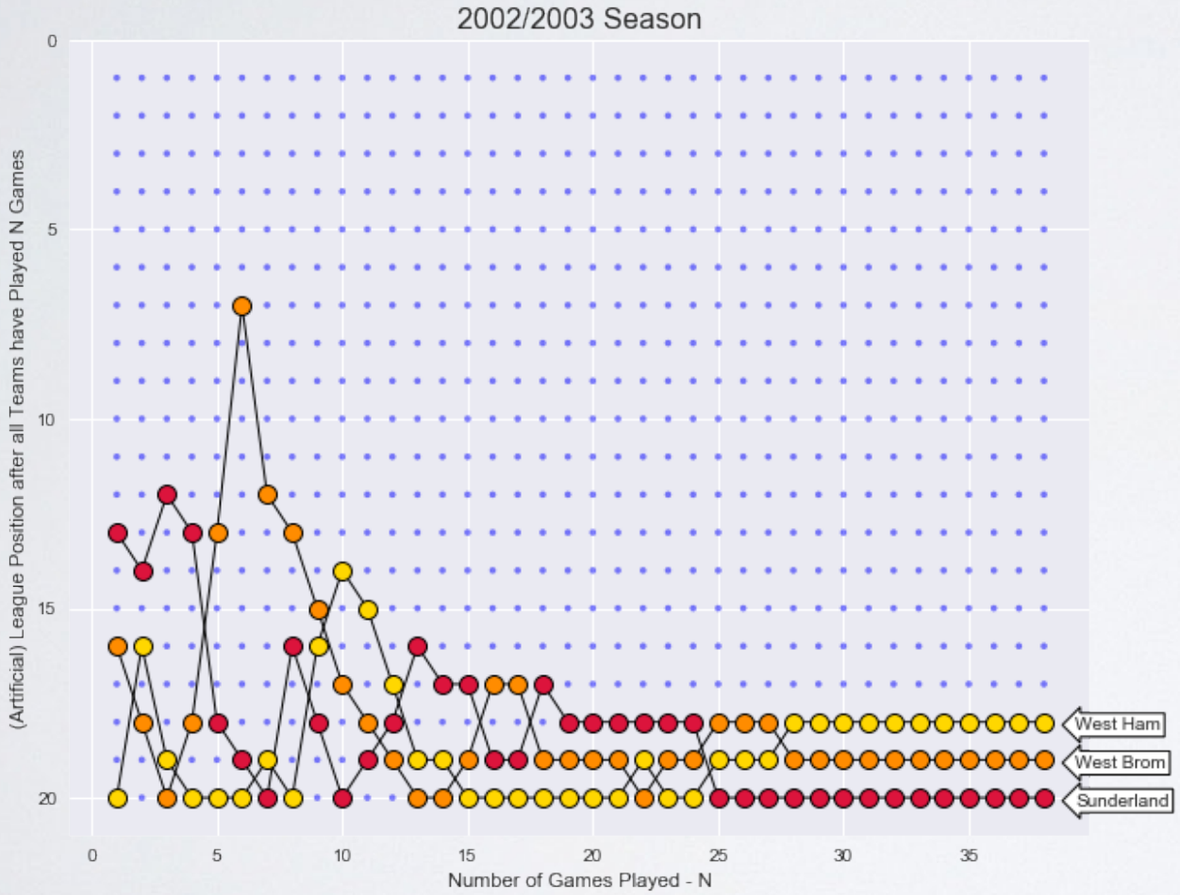
INTERESTING ONLY IF …

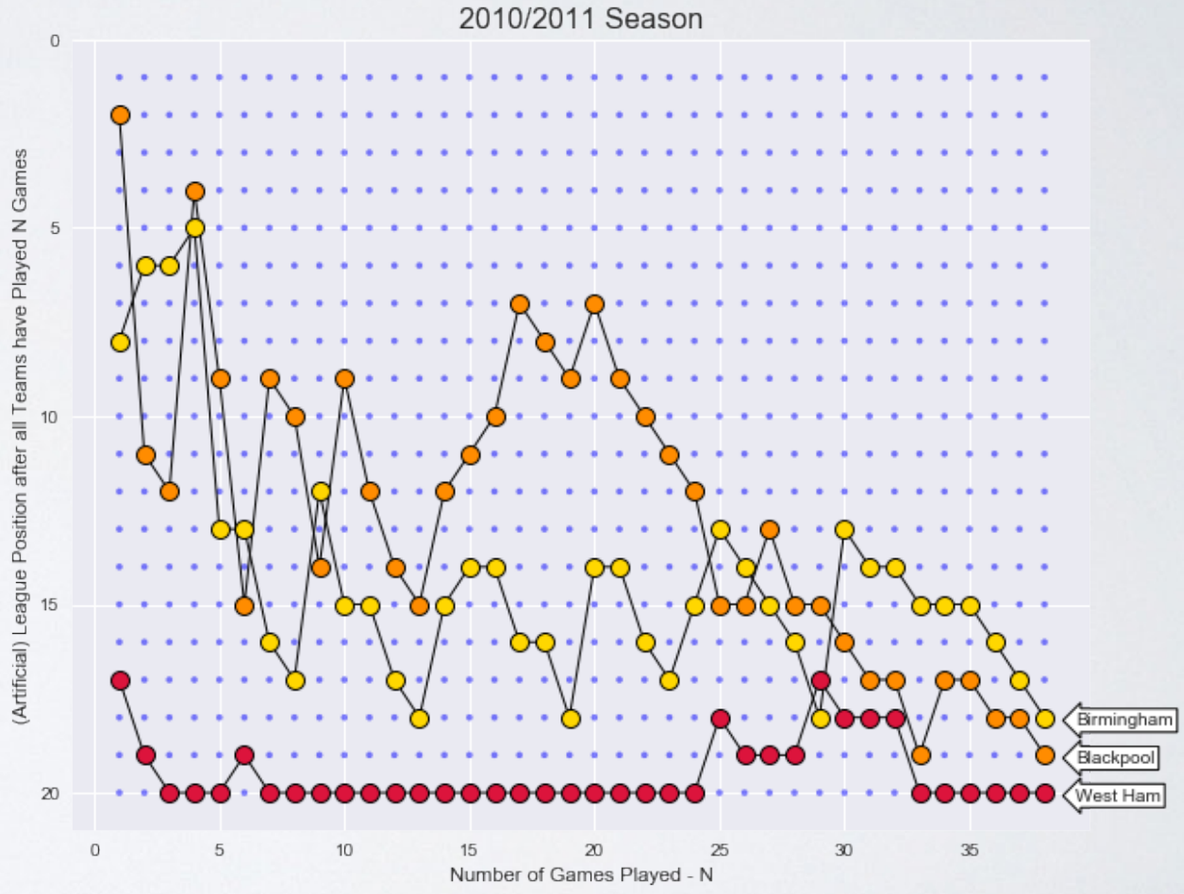
… Teams move in and out of the bottom three.

- There would be no need to model anything if, say, after 10 games, the bottom three consisted of the same three teams until the end of the season

- So what does the journey to relegation look like?

DULL & EXCITING SEASONS





TRAIN TEST SPLIT

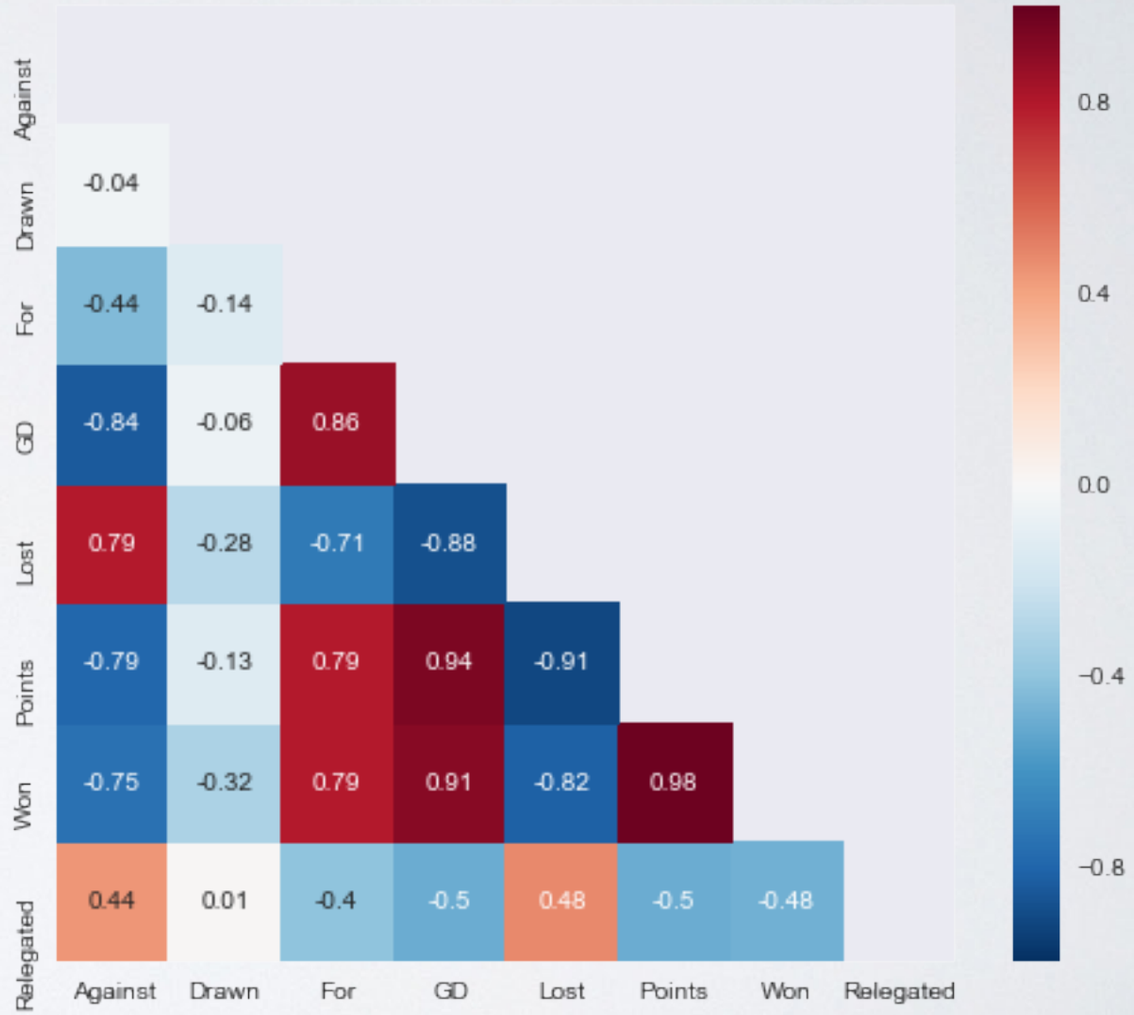
- Manually split data
- Training set to fit the model from seasons 1995/96 to 2009/10 and test set to evaluate the model from 2010/11 to 2016/17
- 300 rows in the training set and 140 in the test set - an approximate 2:1 ratio

CORRELATIONS

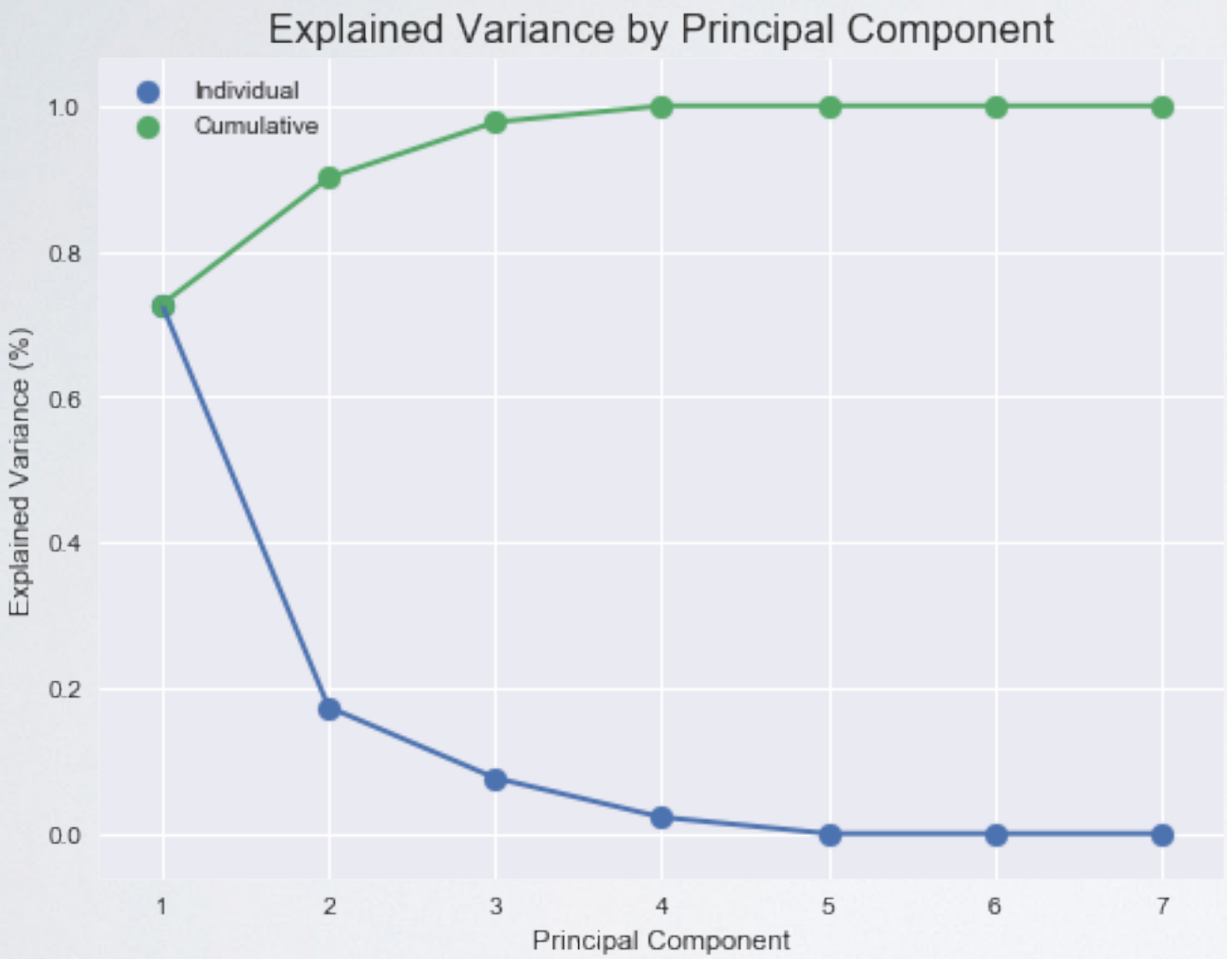
- Important to identify correlations
- Between the target and features to identify predictive power
- Within the features to avoid model overfitting

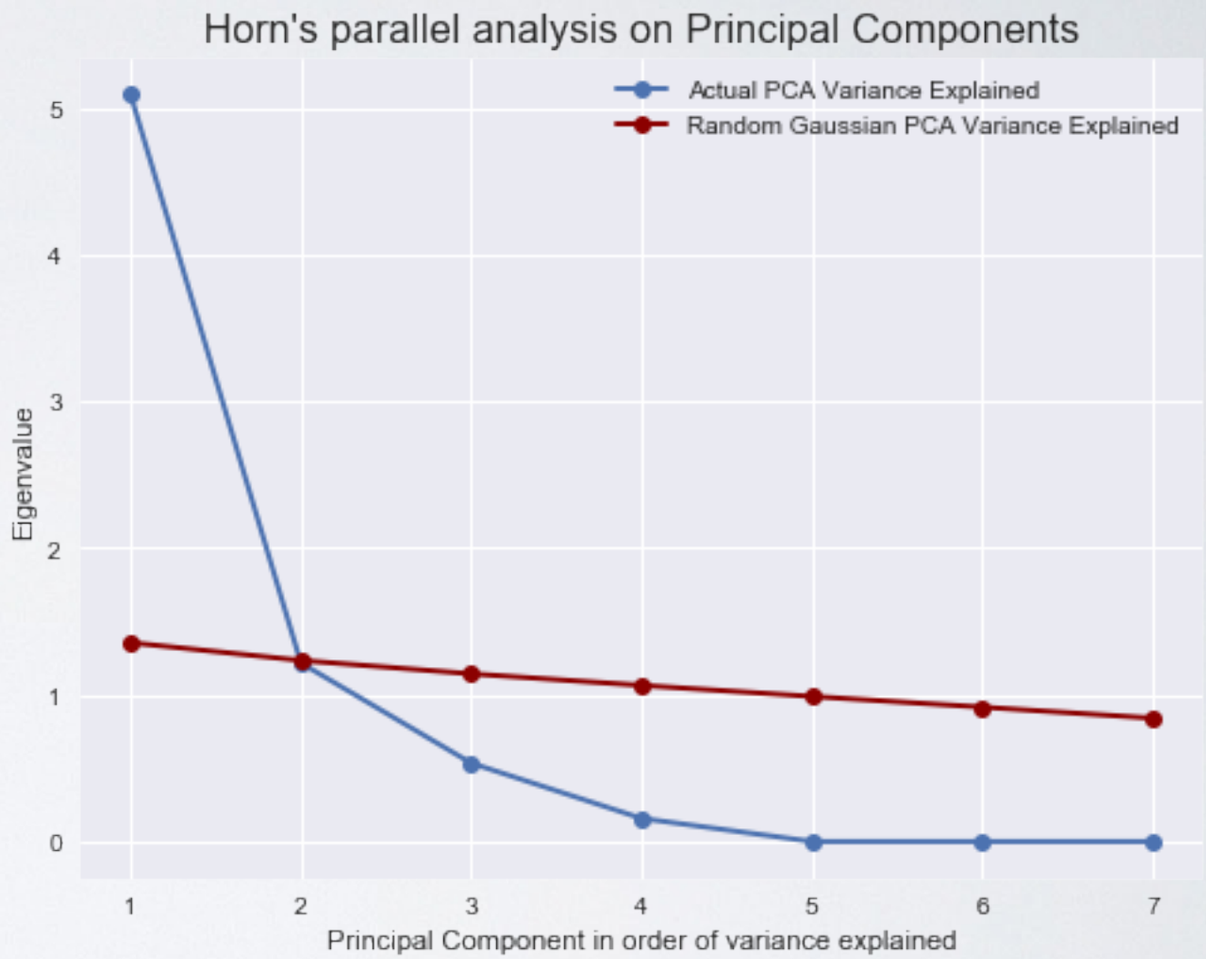
LOTS OF CORRELATIONS

- After 19 games
- Mostly obvious
- Interesting:
  - Games lost and relegation
  - Zero correlation with draws



PCA (AFTER NORMALISATION)





Most of the variance (72%) can be explained by one principal component. This is very approximately the sum of normalised positive features less normalised negative features when it comes to staying up, i.e. 0.4 * (Games Won + Points + Goal Difference + Goals For - Games Lost - Goals Against)

SCORE METRIC

- Baseline = 0.85 accuracy (17/20 teams stay up)
- K-Nearest Neighbours (K=3) = 0.88

```
                predicted_relegated  predicted_not_relegated
relegated       10                   11
not_relegated   6                    113

             precision    recall  f1-score   support

          0       0.91      0.95      0.93       119
          1       0.62      0.48      0.54        21

avg / total       0.87      0.88      0.87       140
```

- For minority classes, recall is a better metric than accuracy. Recall = Proportion of times model predicted relegation for teams actually relegated. 10/21 in example above

- Using the first two PCA features gave a recall of 0.57 compared with 0.48.

- Other models gave similar results

EXPANDING THE FEATURES

Additional Section Two Features

- Number of points won in the n-th game, the (n-1)th game .... (n-4)th game - current form

- Position in the table - 18 and below means relegation

- For the four teams in the two positions above and two positions below, the difference in games played and points - gives context to position

- Historic finishing positions in the previous three seasons

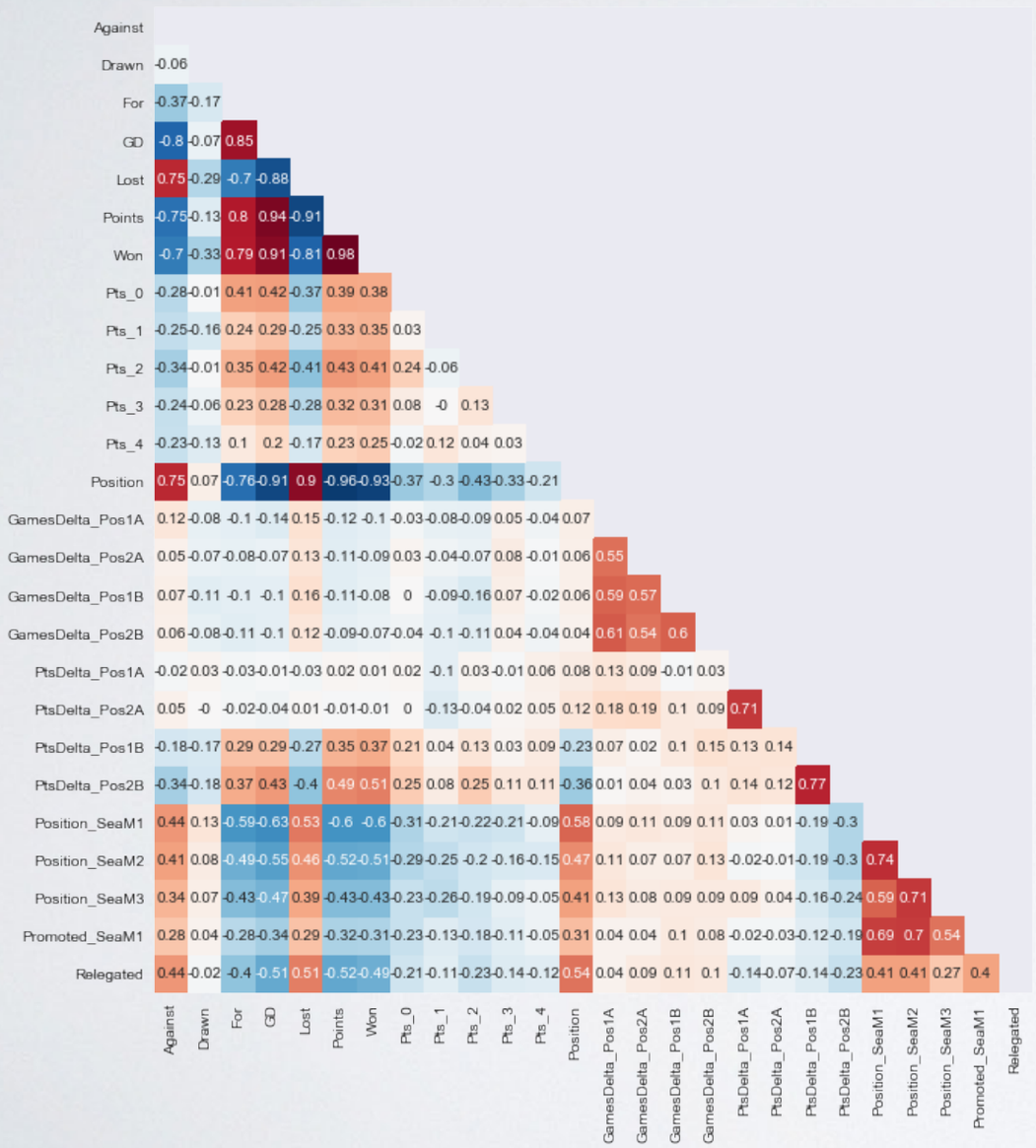
- Promoted team flag

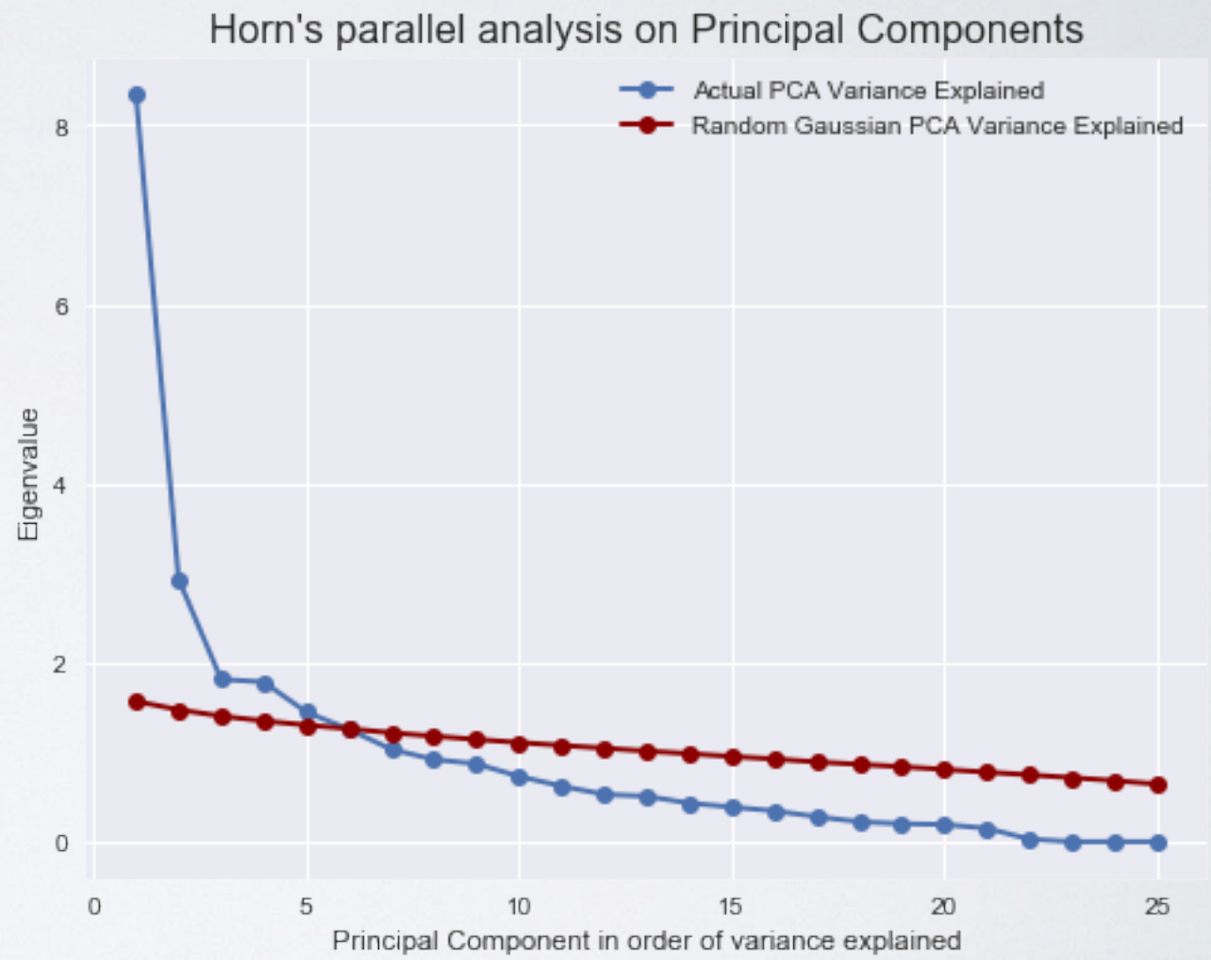
COMPLEX PYTHON CODE

(OK…LONG)

- Over 200 lines of code to generate the features
- After every day of games played, checked to see if any of the features had changed
- Some features depend on the other teams, such as games in hand

EXPANDED FEATURES (AFTER 19 GAMES AGAIN)





RANDOM FOREST

Model: (DEFAULT) RANDOM FOREST

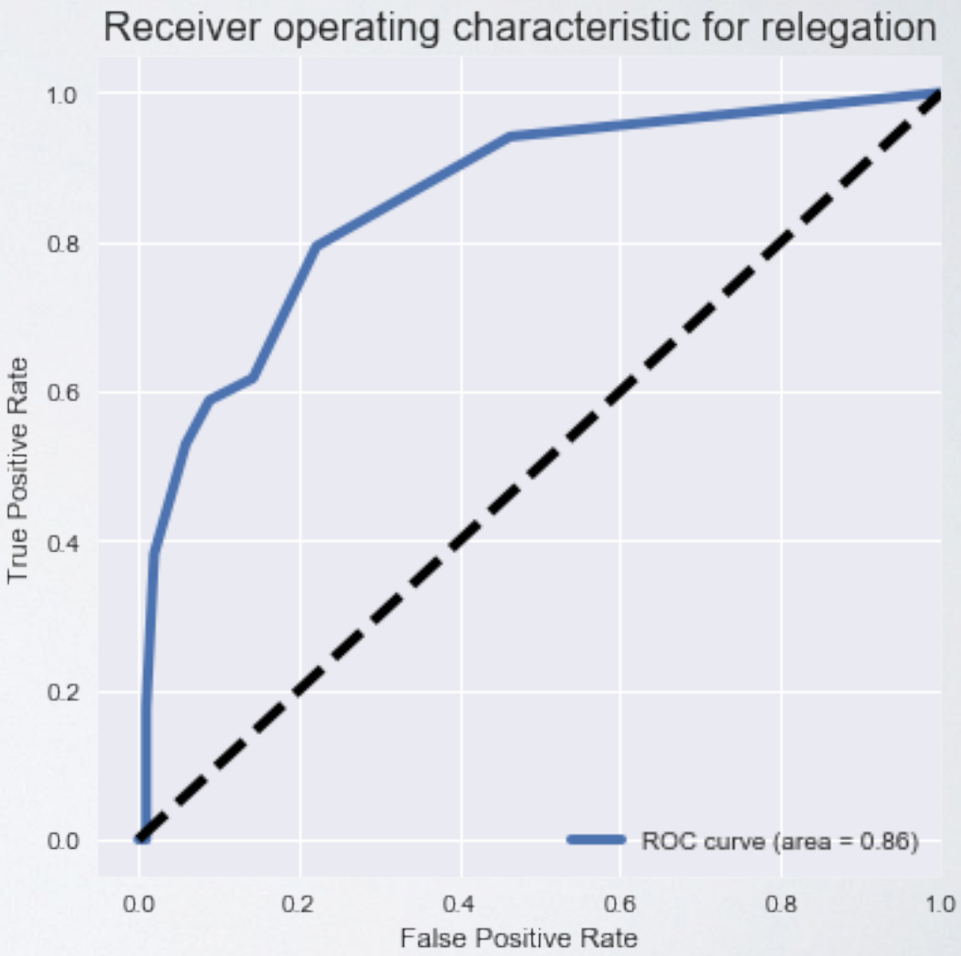
	Predicted Safe	Predicted Relegated	Support
Actual Safe	199	4	203
Actual Relegated	21	13	34
Total Predicted	220	17	237

Baseline accuracy = 0.857
Accuracy = 0.895

	Precision	Recall	F1-Score
Safe	0.905	0.980	0.941
Relegated	0.765	0.382	0.510
Average	0.884	0.895	0.879

- 34 actual teams relegated in 7 seasons ?!?
  This is the features for the 21 actual teams relegated plus 15 more for the same teams but different games in hand information. Still around 15% of all records

- As the probability threshold for relegation increases the true positive rate (recall/sensitivity) increases but so does the false positive rate (1-specificity)



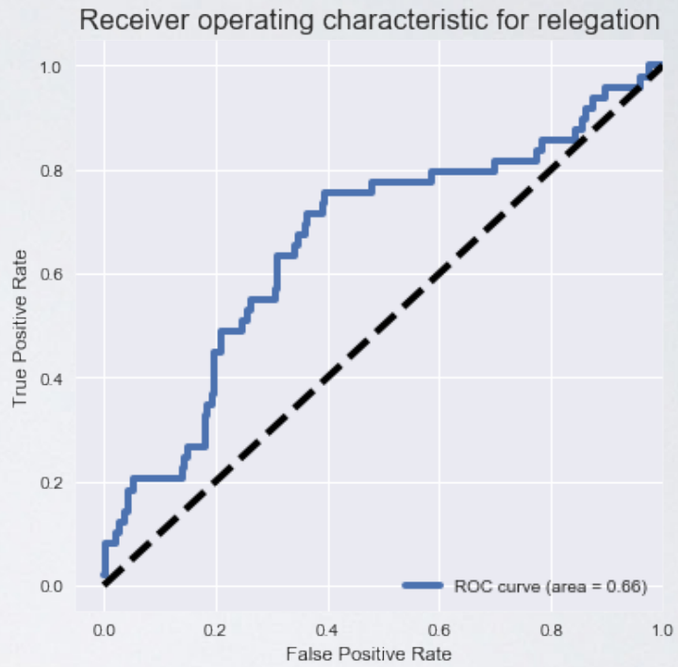
F1-SCORE AND APPLICATION TO SVC

Model: (DEFAULT) SVC

	Predicted Safe	Predicted Relegated	Support
Actual Safe	240	8	248
Actual Relegated	43	6	49
Total Predicted	283	14	297

Baseline accuracy = 0.835
Accuracy = 0.828

	Precision	Recall	F1-Score
Safe	0.848	0.968	0.904
Relegated	0.429	0.122	0.190
Average	0.779	0.828	0.786

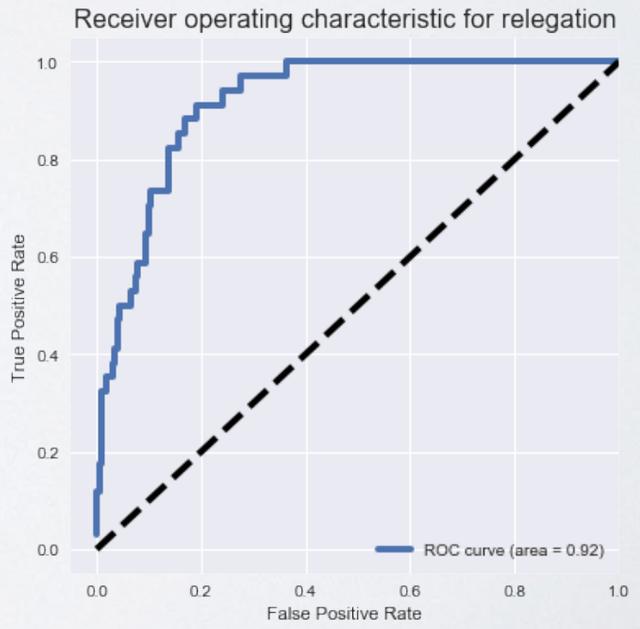


Model: BEST SVC FROM GRIDSEARCH OPTIMISED FOR F1-SCORE

	Predicted Safe	Predicted Relegated	Support
Actual Safe	189	14	203
Actual Relegated	16	18	34
Total Predicted	205	32	237

Baseline accuracy = 0.857
Accuracy = 0.873

	Precision	Recall	F1-Score
Safe	0.922	0.931	0.926
Relegated	0.562	0.529	0.545
Average	0.870	0.873	0.872



BOTTOM THREE MODEL

Model: BOTTOM THREE RELEGATED

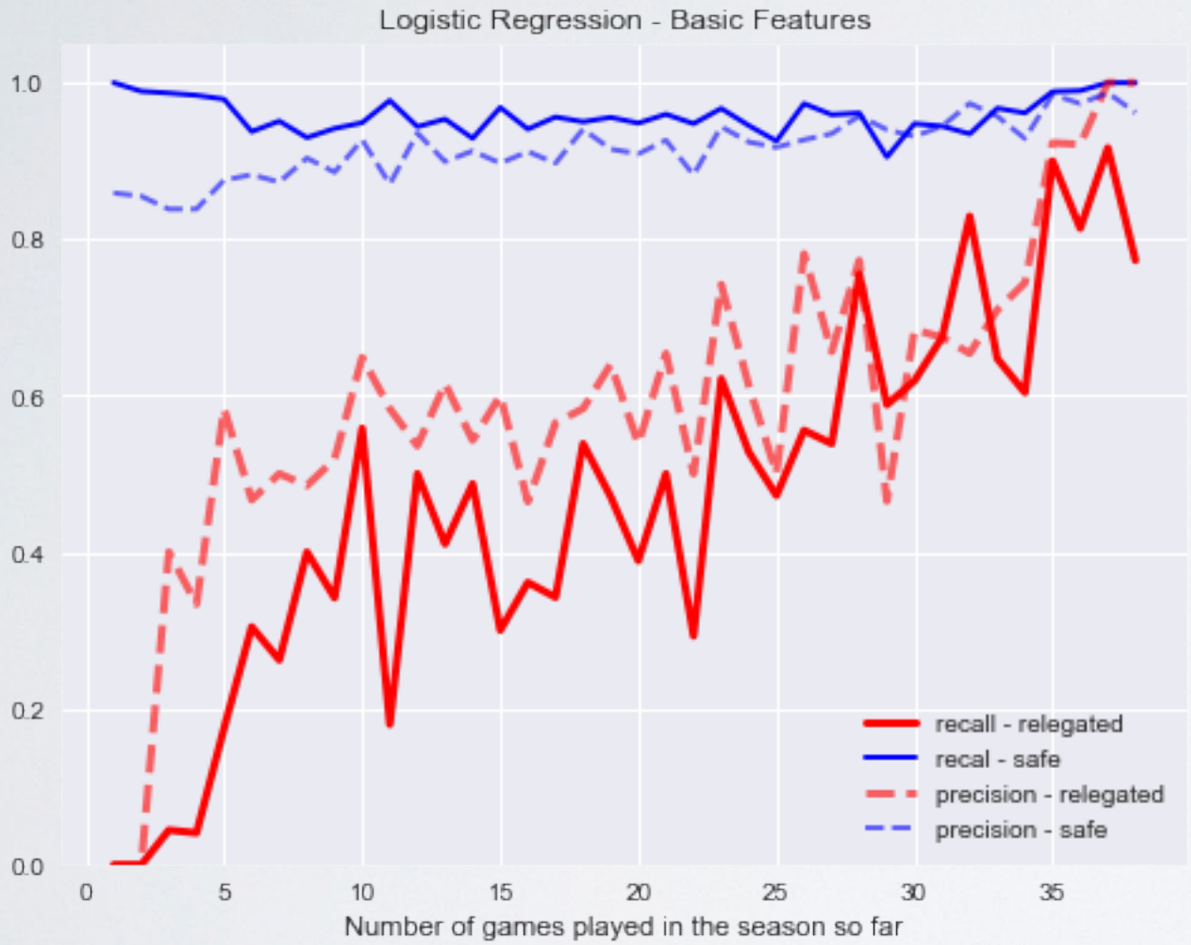
	Predicted Safe	Predicted Relegated	Support
Actual Safe	190	13	203
Actual Relegated	12	22	34
Total Predicted	202	35	237

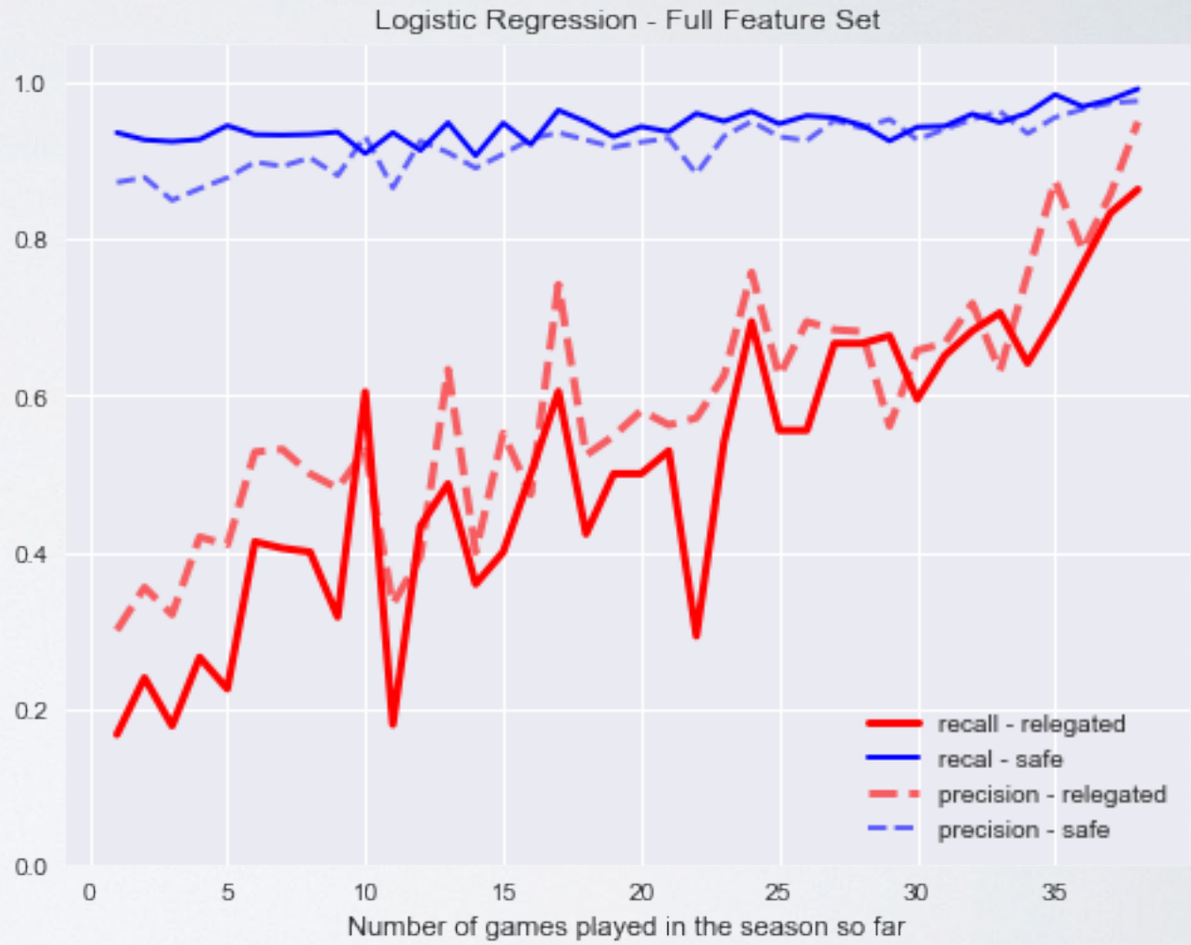
Baseline accuracy = 0.857

Accuracy = 0.895

	Precision	Recall	F1-Score
Safe	0.941	0.936	0.938
Relegated	0.629	0.647	0.638
Average	0.896	0.895	0.895

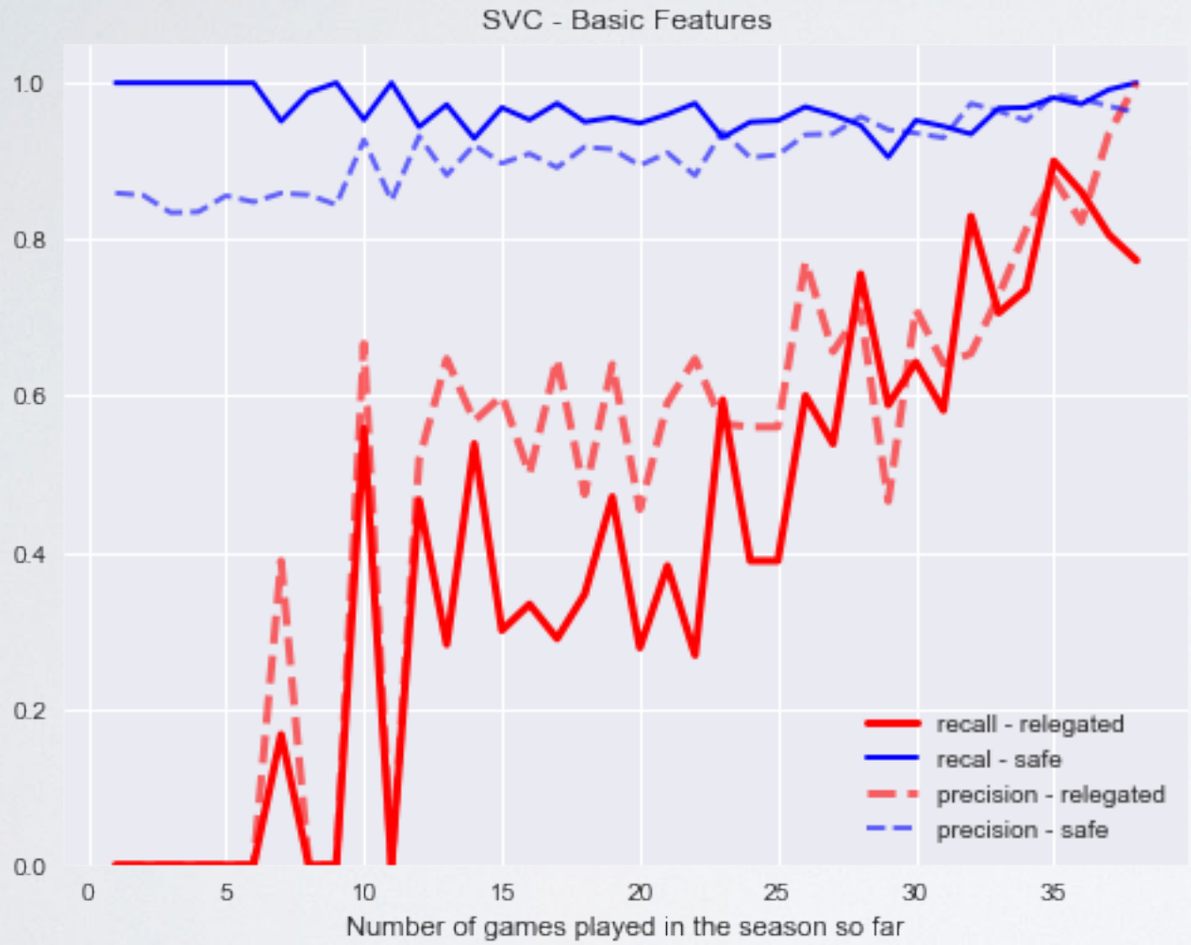
LOGISTIC REGRESSION

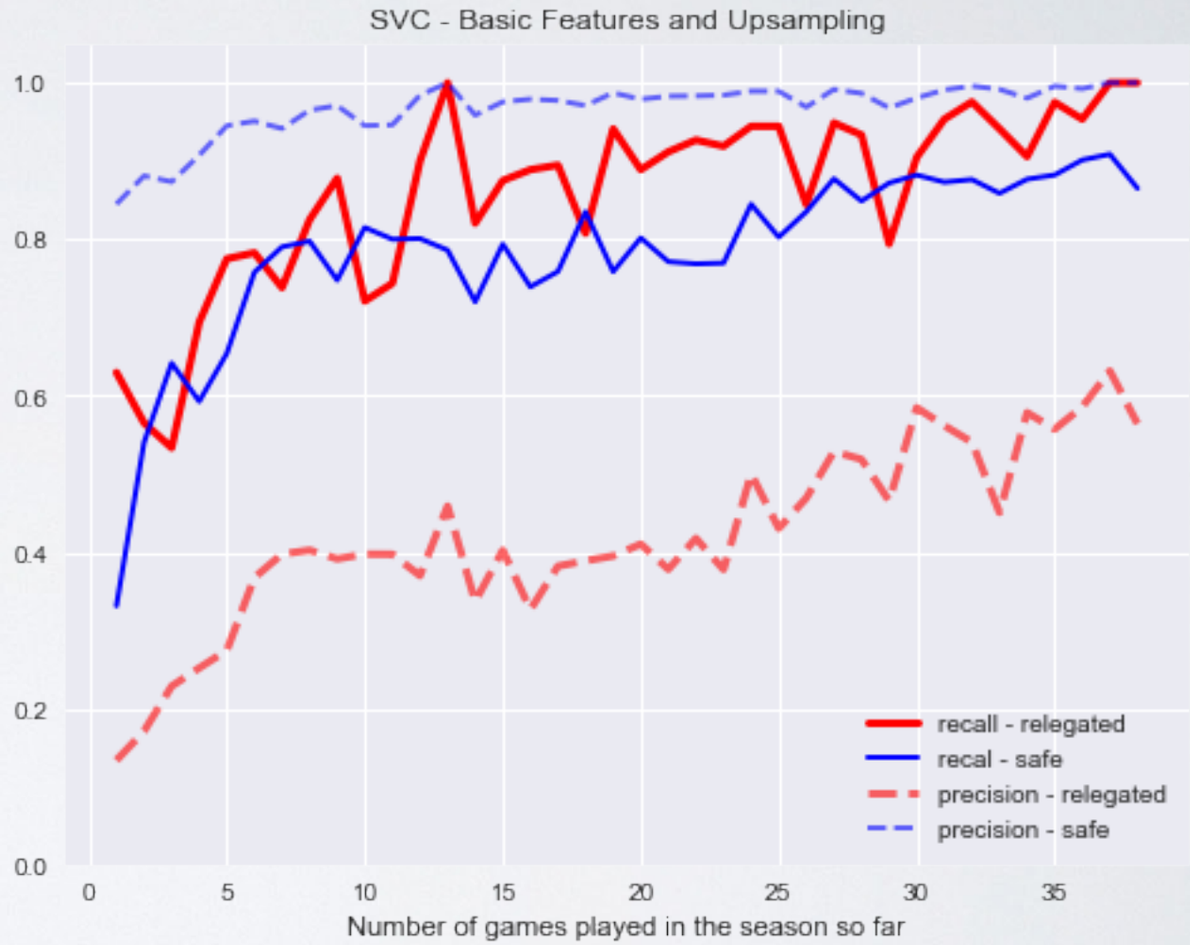




- Adding more features has reduced recall. Need to apply regularisation or PCs

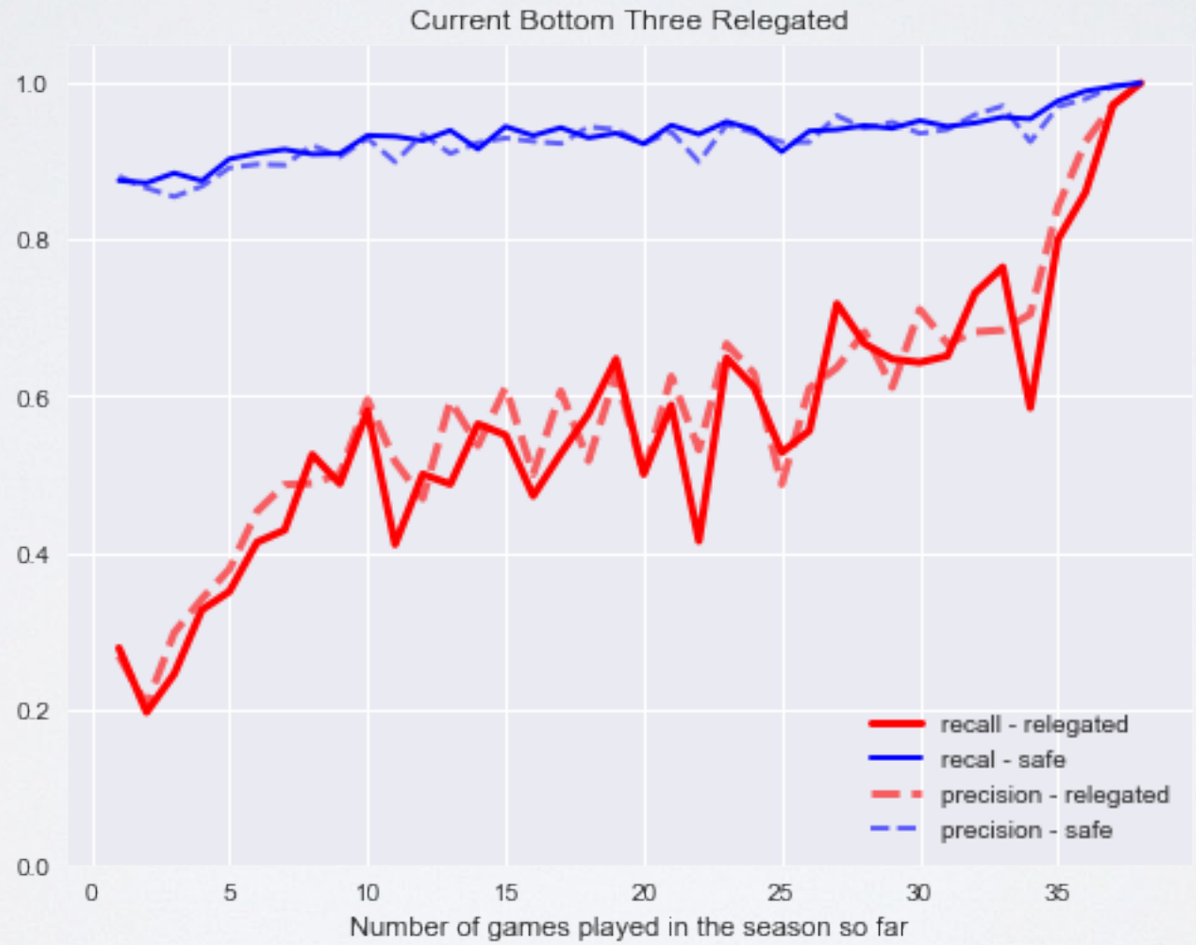
SVC AND UPSAMPLING





- Upsampling has improved recall but increased the false positive rate too

BOTTOM THREE MODEL CHART



CONCLUSION

- Bottom Three Model performs better than ML models

NEXT STEPS / A.K.A. IF I HAD MORE TIME

- Will Bournemouth be relegated this season?
  - BTM says yes. Recall of 0.33 ; Bookies c0.38
- Is it possible to use data from before 1995 in which 3 out of 22 teams were relegated to predict 3 out of 20 teams?
    - Apply the model to all teams in the league. Identify the three teams with the highest probability of relegation and classify these teams as relegated
    - Use a subset of the PCA components rather than the full set and/or regularisation
    - Allow for optimisation of hyperparameters in Section Three.