

العمل في التوقع العلمي في نتائج كرة القدم

عمل وتنفيذ دكتور الدرديري فضل إبراهيم

FOOT-BALL GAMES (PREDICTIONS)

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
import seaborn as sns
```

```
from scipy.stats import poisson,skellam
```

```
import matplotlib.pyplot as py
```

```
from scipy.optimize import minimize
```

```
epl_1718 = pd.read_csv("http://www.football-  
data.co.uk/mmz4281/1718/E0.csv
```

```
[[epl_1718 = epl_1718[['HomeTeam','AwayTeam','FTHG','FTAG
```

```
epl_1718 = epl_1718.rename(columns={'FTHG': 'HomeGoals', 'FTAG':  
{ "AwayGoals
```

```
(epl_1718.head
```

```
AwayGoals HomeGoals AwayTeam HomeTeam
```

```
3    4    Leicester    Arsenal    0
```

```
2    0    Man City    Brighton    1
```

```
3    2    Burnley    Chelsea    2
```

```
3    0    Huddersfield    Crystal Palace    3
```

```
0    1 Stoke    Everton    4
```

```
epl_1718 = epl_1718[:-10]
```

```
(epl_1718.mean
```

```
HomeGoals    1.518919
```

```
AwayGoals    1.148649
```

```
dtype: float64
```

```
,plt.hist(epl_1718[['HomeGoals', 'AwayGoals']].values, range(9
```

```
alpha=0.7, label=['Home', 'Away'],normed=True,
```

```
(["color=["#FFA07A", "#20B2AA
```

```
array([ 0.23783784, 0.33243243, 0.24324324, 0.08918919,  ])
```

```
,0.05945946
```

```
,([ 0.0027027 ,      .0 ,0.03513514
```

```
array([ 0.35675676, 0.33513514, 0.17027027, 0.08918919,
```

```
,0.04054054
```

```
,([      .0 , 0.0027027 ,0.00540541
```

```
,([array([0, 1, 2, 3, 4, 5, 6, 7, 8
```

```
(<a list of 2 Lists of Patches objects>
```

```
poisson_pred = np.column_stack([[poisson.pmf(i, epl_1718.mean()[j])
```

```
([for i in range(8)] for j in range(2
```

```
,[pois1, = plt.plot([i-0.5 for i in range(1,9)], poisson_pred[:,0
```

```
('linestyle='-', marker='o',label="Home", color = '#CD5C5C
```

```
,[pois2, = plt.plot([i-0.5 for i in range(1,9)], poisson_pred[:,1
```

```
('linestyle='-', marker='o',label="Away", color = '#006400
```

```
(leg=plt.legend(loc='upper right', fontsize=13, ncol=2
```

```
leg.set_title("Poisson          Actual          ", prop = {'size':'14',  
({"weight':'bold
```

```
(([plt.xticks([i-0.5 for i in range(1,9)],[i for i in range(9
```

```
,<matplotlib.axis.XTick at 0x20ba0bbedd8>])
```

```
,<matplotlib.axis.XTick at 0x20ba0a94358>
```

```
,<matplotlib.axis.XTick at 0x20ba0c07c88>
```

```
,<matplotlib.axis.XTick at 0x20ba0c94978>
```

```
,<matplotlib.axis.XTick at 0x20ba0c9d080>
```

```
,<matplotlib.axis.XTick at 0x20ba0c9d6d8>
```

```
,<matplotlib.axis.XTick at 0x20ba0c9dcf8>
```

```
,[<matplotlib.axis.XTick at 0x20ba0ca3320>
```

```
(<a list of 8 Text xticklabel objects>
```

```
([skellam.pmf(0.0, epl_1718.mean())[0], epl_1718.mean())[1
```

```
0.25480879523707972
```

```
(plt.xlabel("Goals per Match",size=13
```

```
(plt.ylabel("Proportion of Matches",size=13
```

```
plt.title("Number of Goals per Match (EPL 2016/17  
( 'Season)',size=14,fontweight='bold
```

```
([plt.ylim([-0.004, 0.4
```

```
()plt.tight_layout
```

```
()plt.show
```

```
([skellam.pmf(1, epl_1718.mean())[0], epl_1718.mean()[1]  
0.2284260568855998
```

```
skellam_pred = [skellam.pmf(i, epl_1718.mean())[0],  
[(epl_1718.mean()[1]) for i in range(-6,8
```

```
plt.hist(epl_1718[['HomeGoals']].values -  
,(epl_1718[['AwayGoals']].values, range(-6,8
```

```
(alpha=0.7, label='Actual',normed=True
```

```
,array([ 0.0027027 , 0.        , 0.02162162, 0.05405405, 0.05675676)
```

```
,0.10540541 ,0.21891892 ,0.26756757 ,0.14594595  
,0.07027027
```

```
,([      .0 ,0.02702703 ,0.02972973
```

```
,([array([-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7
```

```
(<a list of 13 Patch objects>
```

```
(fig,(ax1,ax2) = plt.subplots(2, 1
```

```
,plt.plot([i+0.5 for i in range(-6,8)], skellam_pred
```

```
('linestyle='-', marker='o',label="Skellam", color = '#CD5C5C
```

```
(plt.legend(loc='upper right', fontsize=13
```

```
([plt.xticks([i+0.5 for i in range(-6,8)],[i for i in range(-6,8
```

```

(plt.xlabel("Home Goals - Away Goals",size=13
(plt.ylabel("Proportion of Matches",size=13
plt.title("Difference in Goals Scored (Home Team vs Away
('Team)",size=14,fontweight='bold
([plt.ylim([-0.004, 0.26
()plt.tight_layout
()plt.show

```

```

(fig,(ax1,ax2) = plt.subplots(2, 1

```

```

chel_home =
epl_1718[epl_1718['HomeTeam']=='Chelsea']['HomeGoals'].apply(pd.v
(alue_counts,normalize=True

chel_home_pois =
[poisson.pmf(i,np.sum(np.multiply(chel_home.values.T,chel_home.inde
[(x.T),axis=1)[0]) for i in range(8

sun_home =
epl_1718[epl_1718['HomeTeam']=='Sunderland']['HomeGoals'].apply(p
(d.value_counts,normalize=True

sun_home_pois =
[poisson.pmf(i,np.sum(np.multiply(sun_home.values.T,sun_home.index.
[(T),axis=1)[0]) for i in range(8

chel_away =
epl_1718[epl_1718['AwayTeam']=='Chelsea']['AwayGoals'].apply(pd.va
(lue_counts,normalize=True

```

```
chel_away_pois =  
[poisson.pmf(i,np.sum(np.multiply(chel_away.values.T,chel_away.index  
[(.T),axis=1)[0]) for i in range(8
```

```
sun_away =  
epl_1718[epl_1718['AwayTeam']=='Sunderland'][['AwayGoals']].apply(p  
(d.value_counts,normalize=True
```

```
sun_away_pois =  
[poisson.pmf(i,np.sum(np.multiply(sun_away.values.T,sun_away.index.  
[(T),axis=1)[0]) for i in range(8
```

```
import statsmodels.api as sm
```

```
import statsmodels.formula.api as smf
```

```
goal_model_data =  
pd.concat([epl_1718[['HomeTeam','AwayTeam','HomeGoals']].assign(h  
)ome=1).rename
```

```
columns={'HomeTeam':'team',  
,({'AwayTeam':'opponent','HomeGoals':'goals
```

```
epl_1718[['AwayTeam','HomeTeam','AwayGoals']].assign(home=0).ren  
)ame
```

```
columns={'AwayTeam':'team',  
([({'HomeTeam':'opponent','AwayGoals':'goals
```

```
poisson_model = smf.glm(formula="goals ~ home + team + opponent",  
,data=goal_model_data
```

```
()family=sm.families.Poisson()).fit
```

```
(())print(poisson_model.summary
```

Generalized Linear Model Regression Results

```
=====
=====
```

```
Dep. Variable:          goals  No. Observations:          740
Model:                GLM    Df Residuals:              700
Model Family:         Poisson  Df Model:                39
Link Function:         log    Scale:                   1.0
Method:               IRLS    Log-Likelihood:          -1020.2
Date:                 Wed, 26 Dec 2018  Deviance:         768.51
.Time:                01:16:59  Pearson chi2:         653
No. Iterations:         5
```

```
=====
=====
```

```
[coef    std err          z      P>|z|    [0.025    0.975
```

```
-----
```

```
Intercept                0.5855    0.189    3.091    0.002    0.214
0.957

team[T.Bournemouth]      -0.5306    0.193   -2.754    0.006    -
0.908   -0.153

team[T.Brighton]         -0.7740    0.208   -3.721    0.000   -1.182
-0.366

team[T.Burnley]          -0.7418    0.206   -3.602    0.000   -1.145
-0.338

team[T.Chelsea]          -0.1860    0.173   -1.074    0.283    -
0.525    0.153

team[T.Crystal Palace]   -0.5203    0.193   -2.701    0.007    -
0.898   -0.143
```

team[T.Everton] -0.144	-0.5212	0.193	-2.706	0.007	-0.899
team[T.Huddersfield] 1.383 -0.512	-0.9476	0.222	-4.261	0.000	-
team[T.Leicester] 0.011	-0.3458	0.182	-1.901	0.057	-0.702
team[T.Liverpool] 0.400	0.0822	0.162	0.506	0.613	-0.236
team[T.Man City] 0.637	0.3378	0.153	2.211	0.027	0.038
team[T.Man United] 0.431 0.234	-0.0984	0.170	-0.580	0.562	-
team[T.Newcastle] 1.117 -0.318	-0.7174	0.204	-3.517	0.000	-
team[T.Southampton] 1.081 -0.288	-0.6843	0.202	-3.384	0.001	-
team[T.Stoke] -0.369	-0.7811	0.210	-3.716	0.000	-1.193
team[T.Swansea] 1.422 -0.538	-0.9799	0.226	-4.343	0.000	-
team[T.Tottenham] 0.399 0.261	-0.0690	0.168	-0.410	0.682	-
team[T.Watford] -0.132	-0.5066	0.191	-2.648	0.008	-0.881
team[T.West Brom] 1.277 -0.435	-0.8563	0.215	-3.988	0.000	-
team[T.West Ham] 0.832 -0.087	-0.4599	0.190	-2.420	0.016	-
opponent[T.Bournemouth] 0.241 0.508	0.1334	0.191	0.699	0.485	-
opponent[T.Brighton] 0.422 0.360	-0.0313	0.199	-0.157	0.875	-

opponent[T.Burnley]	-0.3603	0.216	-1.666	0.096	-
0.784 0.064					
opponent[T.Chelsea]	-0.3856	0.220	-1.753	0.080	-
0.817 0.045					
opponent[T.Crystal Palace]	0.0399	0.195	0.205	0.838	-
0.342 0.422					
opponent[T.Everton]	0.0539	0.195	0.276	0.782	-
0.328 0.436					
opponent[T.Huddersfield]	0.0789	0.193	0.409	0.683	-
0.299 0.457					
opponent[T.Leicester]	0.0770	0.195	0.395	0.693	-
0.305 0.459					
opponent[T.Liverpool]	-0.2890	0.215	-1.346	0.178	-
0.710 0.132					
opponent[T.Man City]	-0.5969	0.239	-2.503	0.012	-
1.064 -0.129					
opponent[T.Man United]	-0.6041	0.236	-2.564	0.010	-
1.066 -0.142					
opponent[T.Newcastle]	-0.1119	0.203	-0.552	0.581	-
0.509 0.285					
opponent[T.Southampton]	0.0663	0.195	0.341	0.733	-
0.316 0.448					
opponent[T.Stoke]	0.2289	0.186	1.229	0.219	-
0.136 0.594					
opponent[T.Swansea]	0.0055	0.196	0.028	0.977	-
0.378 0.389					
opponent[T.Tottenham]	-0.4647	0.226	-2.056	0.040	-
0.908 -0.022					
opponent[T.Watford]	0.2030	0.189	1.075	0.282	-
0.167 0.573					
opponent[T.West Brom]	0.0204	0.196	0.104	0.917	-
0.363 0.404					

opponent[T.West Ham]	0.2460	0.186	1.320	0.187	-
0.119	0.611				
home	0.2792	0.064	4.339	0.000	0.153
0.405					

```
=====
=====
```

```
poisson_model.predict(pd.DataFrame(data={'team': 'Arsenal',
,"opponent": 'Southampton
```

```
(([home':1},index=[1'
```

```
2.537017 1
```

```
dtype: float64
```

```
poisson_model.predict(pd.DataFrame(data={'team': 'Southampton',
,"opponent": 'Arsenal
```

```
(([home':0},index=[1'
```

```
0.905874 1
```

```
dtype: float64
```

```
def simulate_match(foot_model, homeTeam, awayTeam,
:(max_goals=10
```

```
home_goals_avg = foot_model.predict(pd.DataFrame(data={'team':
,homeTeam
```

```
,{'opponent': awayTeam,'home':1'
```

```
[index=[1])).values[0
```

```
away_goals_avg = foot_model.predict(pd.DataFrame(data={'team':
,awayTeam
```

```
,{'opponent': homeTeam,'home':0'
```

```
[index=[1])).values[0
```

```

team_pred = [[poisson.pmf(i, team_avg) for i in range(0,
[[max_goals+1)]] for team_avg in [home_goals_avg, away_goals_avg

(((return(np.outer(np.array(team_pred[0]), np.array(team_pred[1

ars_sou = simulate_match(poisson_model, 'Arsenal', 'Southampton',
(max_goals=10

([print(ars_sou[0:5, 0:5

[0.00089708 0.00396117 0.01311828 0.02896271 0.03197214 ]]

[ 0.0022759 0.01004955 0.03328129 0.07347889 0.08111385 ]

[ 0.002887 0.01274794 0.0422176 0.0932086 0.10289361 ]

[0.00244146 0.01078058 0.03570226 0.07882393 0.08701428 ]

[[0.00154851 0.00683763 0.02264431 0.04999441 0.05518917 ]

```

```

from matplotlib.colors import ListedColormap

```

```

def matrix_gif(matrix, colour_matrix, colour_map, subtitle="",
:(heatmap=False, alpha=0.8

((fig, ax1 = plt.subplots(1, figsize=(5,5

:if heatmap

(ax1.matshow(matrix, alpha=alpha

:else

(ax1.matshow(colour_matrix, cmap=colour_map, alpha=alpha

(ax1.tick_params(axis=u'both', which=u'both',length=0

("=ax1.grid(which='major', axis='both', linestyle

(ax1.set_xlabel('Away Team Goals', fontsize=12

(ax1.set_ylabel('Home Team Goals', fontsize=12

('ax1.xaxis.set_label_position('top

nrows, ncols = matrix.shape

```

```

:(for i in range(nrows
:(for j in range(ncols
[c = matrix[i][j]
(ax1.text(j, i, str(round(c,4)), va='center', ha='center', size=13
,plt.figtext(0.5, 0.05, subtitle, horizontalalignment='center
('fontsize=14, multialignment='left', fontweight='bold
return fig

(['cmap = ListedColormap(['w', '#04f5ff', '#00ff85', '#e90052
matrix = simulate_match(poisson_model, 'Arsenal', 'Southampton',
(max_goals=5
(matn = len(matrix
,matrix_gif(matrix, matrix, ListedColormap(['w']), heatmap=True
alpha=0.6, subtitle="Match Score Probability
("Matrix").savefig("match_matrix_0.png
())plt.close

for t,(mat,colour,subtitle) in enumerate(zip([np.zeros((matn, matn)),
,(np.tril(np.ones((matn,matn)),-1
,(np.triu(np.ones((matn,matn))*2,1), np.diag([3]*matn
np.array([0 if i+j<3 else 1 for i in
,(range(matn) for j in range(matn))]).reshape(matn,matn
,['w', '#04f5ff', '#00ff85', '#e90052','#EAF205']
Match Score Probability Matrix', 'Home Win', '']
:([["Away Win", 'Draw', 'Over 2.5 goals
matrix_gif(matrix, mat, ListedColormap(['w'] + [colour]),
,heatmap=False
alpha=0.6,
((subtitle=subtitle).savefig("match_matrix_{}.png".format(t+1

```

```
()plt.show
```

```
:(def poiss_actual_diff(football_url, max_goals
(epl_1718 = pd.read_csv(football_url
[['epl_1718 = epl_1718[['HomeTeam','AwayTeam','FTHG','FTAG
epl_1718 = epl_1718.rename(columns={'FTHG': 'HomeGoals',
({'FTAG': 'AwayGoals
(team_pred = [[poisson.pmf(i, team_avg) for i in range(0, max_goals
\
for team_avg in [epl_1718['HomeGoals'].mean(),
[()epl_1718['AwayGoals'].mean
\ - ([return np.outer(np.array(team_pred[0]), np.array(team_pred[1
np.array([sum((epl_1718['HomeGoals']==i) &
(((epl_1718['AwayGoals']==j
for i in range(max_goals) for j in
(range(max_goals))).reshape((6,6))/len(epl_1718
```

```

[] = year_arrays

:(for year in range(2005,2018

year_arrays.append(poiss_actual_diff("http://www.football-
)data.co.uk/mmz4281/{}/E0.csv".format

((str(year)[-2:], str(year+1)[-2:]),6

(cmap = sns.diverging_palette(10, 133, as_cmap=True

((fig, ax = plt.subplots(figsize=(5,5

:("with sns.axes_style("white

ax = sns.heatmap(np.mean(year_arrays, axis=0), annot=True,
,fmt='.4f', cmap=cmap, vmin=-0.013, vmax=.013, center=0.00

square=True, linewidths=.5, annot_kws={"size": 11},
({cbar_kws={"shrink": .8

(ax.tick_params(axis=u'both', which=u'both',length=0

("=ax.grid(which='major', axis='both', linestyle

(ax.set_xlabel('Away Team Goals', fontsize=13

(ax.set_ylabel('Home Team Goals', fontsize=13

('ax.xaxis.set_label_position('top

('ax.xaxis.set_ticks_position('top

plt.figtext(0.45, 0.1, 'Actual Proportion - Model Probability',
,'horizontalalignment='center

('fontsize=14, multialignment='left', fontweight='bold

())plt.tight_layout

())plt.show

```

```

:(def rho_correction(x, y, lambda_x, mu_y, rho
:if x==0 and y==0
(return 1- (lambda_x * mu_y * rho
:elif x==0 and y==1
(return 1 + (lambda_x * rho
:elif x==1 and y==0
(return 1 + (mu_y * rho
:elif x==1 and y==1
return 1 - rho
:else
return 1.0

```

```

:(def dc_log_like(x, y, alpha_x, beta_x, alpha_y, beta_y, rho, gamma
lambda_x, mu_y = np.exp(alpha_x + beta_y + gamma),
(np.exp(alpha_y + beta_x
+ ((return (np.log(rho_correction(x, y, lambda_x, mu_y, rho
np.log(poisson.pmf(x, lambda_x)) + np.log(poisson.pmf(y,
(((mu_y

```

```

def solve_parameters(dataset, debug = False, init_vals=None,
, {options={'disp': True, 'maxiter':100
constraints = [{'type':'eq', 'fun': lambda x: sum(x[:20])-20}] ,
: (**kwargs
(())teams = np.sort(dataset['HomeTeam'].unique
check for no weirdness in dataset #
(())away_teams = np.sort(dataset['AwayTeam'].unique

```

```

:(if not np.array_equal(teams, away_teams
("raise ValueError("Something's not right
(n_teams = len(teams
:if init_vals is None
random initialisation of model parameters #
init_vals = np.concatenate((np.random.uniform(0,1,(n_teams)), #
attack strength
np.random.uniform(0,-1,(n_teams)), # defence
strength
np.array([0, 1.0]) # rho (score correction),
(gamma (home advantage
((
:(def dc_log_like(x, y, alpha_x, beta_x, alpha_y, beta_y, rho, gamma
lambda_x, mu_y = np.exp(alpha_x + beta_y + gamma),
(np.exp(alpha_y + beta_x
+ ((return (np.log(rho_correction(x, y, lambda_x, mu_y, rho
np.log(poisson.pmf(x, lambda_x)) + np.log(poisson.pmf(y,
(((mu_y
:(def estimate_paramters(params
([score_coefs = dict(zip(teams, params[:n_teams
([(defend_coefs = dict(zip(teams, params[n_teams:(2*n_teams
[:rho, gamma = params[-2
log_like = [dc_log_like(row.HomeGoals, row.AwayGoals,
,score_coefs[row.HomeTeam], defend_coefs[row.HomeTeam
score_coefs[row.AwayTeam],
defend_coefs[row.AwayTeam], rho, gamma) for row in
[()]dataset.itertuples
(return -sum(log_like

```



```
opt_output = minimize(estimate_paramters, init_vals,  
(options=options, constraints = constraints, **kwargs
```

```
:if debug
```

```
sort of hacky way to investigate the output of the optimisation #  
process
```

```
return opt_output
```

```
:else
```

```
+ [return dict(zip(["attack_" + team for team in teams
```

```
+ ["defence_" + team for team in teams"]
```

```
,['rho', 'home_adv']
```

```
((opt_output.x
```

```
(params = solve_parameters(epl_1718
```

```
C:\Users\DIRDIR\AppData\Local\Continuum\anaconda3\lib\site-  
packages\ipykernel_launcher.py:18: RuntimeWarning: divide by zero  
encountered in log
```

```
C:\Users\DIRDIR\AppData\Local\Continuum\anaconda3\lib\site-  
packages\ipykernel_launcher.py:18: RuntimeWarning: invalid value  
encountered in log
```

```
(Optimization terminated successfully. (Exit mode 0
```

```
Current function value: 1018.2426976476618
```

```
Iterations: 56
```

```
Function evaluations: 2550
```

```
Gradient evaluations: 56
```

```
params
```

```
,attack_Arsenal': 1.4648142298480571'}
```

```
,attack_Bournemouth': 0.93113641857197105'
```

,attack_Brighton': 0.70694340992857407'
,attack_Burnley': 0.70605333209736354'
,attack_Chelsea': 1.278105977670855'
,attack_Crystal Palace': 0.94096897228372833'
,attack_Everton': 0.94842474860340364'
,attack_Huddersfield': 0.53412121559506731'
,attack_Leicester': 1.1329459072887267'
,attack_Liverpool': 1.5514657210419127'
,attack_Man City': 1.8072646940822659'
,attack_Man United': 1.3577410549133089'
,attack_Newcastle': 0.70833924593956421'
,attack_Southampton': 0.79261324548106593'
,attack_Stoke': 0.69025628983131071'
,attack_Swansea': 0.47027703829482476'
,attack_Tottenham': 1.391314706341928'
,attack_Watford': 0.95342929365886531'
,attack_West Brom': 0.61172499584256623'
,attack_West Ham': 1.0220595026846429'
,defence_Arsenal': -0.87841529803336293'
,defence_Bournemouth': -0.75049065773546941'
,defence_Brighton': -0.91734945989907735'
,defence_Burnley': -1.2558876859373314'
,defence_Chelsea': -1.2759760496035195'
,defence_Crystal Palace': -0.83340671172432856'
,defence_Everton': -0.82799108800611176'
,defence_Huddersfield': -0.80377021526267012'

,defence_Leicester': -0.796836258660654'
,defence_Liverpool': -1.1534397907896747'
,defence_Man City': -1.4807174326196608'
,defence_Man United': -1.4878945411081537'
,defence_Newcastle': -1.012445183194113'
,defence_Southampton': -0.81379964067196986'
,defence_Stoke': -0.65840192921987362'
,defence_Swansea': -0.88267148954438734'
,defence_Tottenham': -1.3648071533103838'
,defence_Watford': -0.67810524317000365'
,defence_West Brom': -0.8765267745724804'
,defence_West Ham': -0.63415373318843293'
,home_adv': 0.2860381283717664'
{rho': -0.14562632628323352'

(params = solve_parameters(epl_1718

C:\Users\DIRDIR\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launcher.py:18: RuntimeWarning: divide by zero encountered in log

C:\Users\DIRDIR\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launcher.py:18: RuntimeWarning: invalid value encountered in log

(Optimization terminated successfully. (Exit mode 0

Current function value: 1018.2426976767588

Iterations: 55

Function evaluations: 2502

Gradient evaluations: 55

params

,attack_Arsenal': 1.4648142298480571'}
,attack_Bournemouth': 0.93113641857197105'
,attack_Brighton': 0.70694340992857407'
,attack_Burnley': 0.70605333209736354'
,attack_Chelsea': 1.278105977670855'
,attack_Crystal Palace': 0.94096897228372833'
,attack_Everton': 0.94842474860340364'
,attack_Huddersfield': 0.53412121559506731'
,attack_Leicester': 1.1329459072887267'
,attack_Liverpool': 1.5514657210419127'
,attack_Man City': 1.8072646940822659'
,attack_Man United': 1.3577410549133089'
,attack_Newcastle': 0.70833924593956421'
,attack_Southampton': 0.79261324548106593'
,attack_Stoke': 0.69025628983131071'
,attack_Swansea': 0.47027703829482476'
,attack_Tottenham': 1.391314706341928'
,attack_Watford': 0.95342929365886531'
,attack_West Brom': 0.61172499584256623'
,attack_West Ham': 1.0220595026846429'
,defence_Arsenal': -0.87841529803336293'
,defence_Bournemouth': -0.75049065773546941'
,defence_Brighton': -0.91734945989907735'
,defence_Burnley': -1.2558876859373314'

```
,defence_Chelsea': -1.2759760496035195'
,defence_Crystal Palace': -0.83340671172432856'
,defence_Everton': -0.82799108800611176'
,defence_Huddersfield': -0.80377021526267012'
,defence_Leicester': -0.796836258660654'
,defence_Liverpool': -1.1534397907896747'
,defence_Man City': -1.4807174326196608'
,defence_Man United': -1.4878945411081537'
,defence_Newcastle': -1.012445183194113'
,defence_Southampton': -0.81379964067196986'
,defence_Stoke': -0.65840192921987362'
,defence_Swansea': -0.88267148954438734'
,defence_Tottenham': -1.3648071533103838'
,defence_Watford': -0.67810524317000365'
,defence_West Brom': -0.8765267745724804'
,defence_West Ham': -0.63415373318843293'
,home_adv': 0.2860381283717664'
{rho': -0.14562632628323352'
```

```
:(def calc_means(param_dict, homeTeam, awayTeam
return [np.exp(param_dict['attack_'+homeTeam] +
,['param_dict['defence_'+awayTeam] + param_dict['home_adv
np.exp(param_dict['defence_'+homeTeam] +
[['param_dict['attack_'+awayTeam
```

```
def dixon_coles_simulate_match(params_dict, homeTeam, awayTeam,
:(max_goals=10
```

```

(team_avgs = calc_means(params_dict, homeTeam, awayTeam

team_pred = [[poisson.pmf(i, team_avg) for i in range(0,
[max_goals+1])] for team_avg in team_avgs

output_matrix = np.outer(np.array(team_pred[0]),
([np.array(team_pred[1]

correction_matrix = np.array([[rho_correction(home_goals,
,[away_goals, team_avgs[0

team_avgs[1], params['rho']]) for
[away_goals in range(2

([for home_goals in range(2

output_matrix[:2,:2] = output_matrix[:2,:2] * correction_matrix

return output_matrix

```

```

ars_sou_dc = dixon_coles_simulate_match(params, 'Arsenal',
('Southampton', max_goals=10

```

```

[Simple Poisson, Dixon-Coles] #

```

```

(print("Arsenal Win

```

```

print('; '.join("{0}: {1:.5f}".format(model, prob) for model,prob in

```

```

zip(["Basic Poisson", "Dixon-Coles"], list(map(lambda
((((([x:np.sum(np.tril(x, -1)), [ars_sou, ars_sou_dc

```

```

(print("Southampton Win

```

```

print('; '.join("{0}: {1:.5f}".format(model, prob) for model,prob in

```

```

zip(["Basic Poisson", "Dixon-Coles"], list(map(lambda
((((([x:np.sum(np.triu(x, 1)), [ars_sou, ars_sou_dc

```

```

(print("Draw

```

```

print('; '.join("{0}: {1:.5f}".format(model, prob) for model,prob in

```

```

zip(["Basic Poisson", "Dixon-Coles"], list(map(lambda
((((([x:np.sum(np.diag(x)), [ars_sou, ars_sou_dc

```

Arsenal Win

Basic Poisson: 0.72700; Dixon-Coles: 0.71626

Southampton Win

Basic Poisson: 0.11278; Dixon-Coles: 0.10287

Draw

Basic Poisson: 0.16015; Dixon-Coles: 0.18079

```
(cmap = sns.diverging_palette(10, 133, as_cmap=True
```

```
((fig, ax = plt.subplots(figsize=(5,5
```

```
:("with sns.axes_style("white
```

```
ax = sns.heatmap(simulate_match(poisson_model, 'Arsenal',  
 \ - ('Southampton', max_goals=5
```

```
dixon_coles_simulate_match(params, 'Arsenal',  
 ,('Southampton', max_goals=5
```

```
annot=True, fmt='.4f', cmap=cmap, vmin=-0.013,  
 ,vmax=.013, center=0.00
```

```
square=True, linewidths=.5, annot_kws={"size": 11},  
 ({cbar_kws={"shrink": .8
```

```
(ax.tick_params(axis=u'both', which=u'both',length=0
```

```
("=ax.grid(which='major', axis='both', linestyle
```

```
(ax.set_xlabel('Away Team Goals', fontsize=13
```

```
(ax.set_ylabel('Home Team Goals', fontsize=13
```

```
('ax.xaxis.set_label_position('top
```

```
('ax.xaxis.set_ticks_position('top
```

```
plt.figtext(0.45, 0.07, ' BP Probs - DC Probs \nArsenal v  
, 'Southampton', horizontalalignment='center
```

```
('fontsize=14, multialignment='left', fontweight='bold
```

```
()plt.tight_layout
```

```
()plt.show
```

```
((fig,(ax1,ax2) = plt.subplots(2, 1, figsize=(10,5
```

```
ax1.plot(range(1000), [0 if y >600 else 1 for y in range(1000)],  
("=label='Component 1', color='#38003c', marker
```

```
ax2.plot(range(1000), np.exp([y*-0.005 for y in range(1000)]),  
("=label='Component 1', color='#07F2F2', marker
```

```
ax2.plot(range(1000), np.exp([y*-0.003 for y in range(1000)]),  
("=label='Component 1', color='#05F26C', marker
```

```
ax2.plot(range(1000), np.exp([y*-0.001 for y in range(1000)]),  
("=label='Component 1', color='#e90052', marker
```

```
([ax1.set_ylim([-0.05,1.05
```

```
([ax2.set_ylim([-0.05,1.05
```

```
([ax1.set_xlim([-0.5,1000
```

```
([ax2.set_xlim([-0.5,1000
```

```
([])ax1.set_xticklabels
```

```
(ax2.xaxis.set_tick_params(labels=12
```

```
(ax1.yaxis.set_tick_params(labels=12
```

```
(ax2.yaxis.set_tick_params(labels=12
```

```
ax1.set_title("Time Decay Weighting  
'Functions",size=14,fontweight='bold
```

```
(ax2.set_xlabel("Number of Days Ago",size=13
```



```

(ax1.set_ylabel(" $\phi(t)$ ",size=13
(ax2.set_ylabel(" $\phi(t)$ ",size=13
ax1.text(830, 0.5, '1  $t \leq \text{\texttt{\textbackslash mathregular\{t\_0\}}n0$   $t >$ 
, $\text{\texttt{\textbackslash mathregular\{t\_0$ 
,'verticalalignment='bottom', horizontalalignment='left
(color='black', fontsize=15
,}' ,ax1.text(800, 0.5
,'verticalalignment='bottom', horizontalalignment='left
(color='black', fontsize=44
,' = (ax1.text(730, 0.62, ' $\phi(t$ 
,'verticalalignment='bottom', horizontalalignment='left
(color='black', fontsize=15
,'(ax2.text(730, 0.62, ' $\phi(t) = \exp(-\xi t$ 
,'verticalalignment='bottom', horizontalalignment='left
(color='black', fontsize=15
,'ax2.text(250, 0.8, ' $\xi = 0.001$ 
,'verticalalignment='bottom', horizontalalignment='left
(color='#e90052', fontsize=15
,'ax2.text(250, 0.5, ' $\xi = 0.003$ 
,'verticalalignment='bottom', horizontalalignment='left
(color='#05F26C', fontsize=15
,'ax2.text(250, 0.0, ' $\xi = 0.005$ 
,'verticalalignment='bottom', horizontalalignment='left
(color='#07F2F2', fontsize=15
())plt.tight_layout
()plt.show

```

```

def dc_log_like_decay(x, y, alpha_x, beta_x, alpha_y, beta_y, rho,
:(gamma, t, xi=0

lambda_x, mu_y = np.exp(alpha_x + beta_y + gamma),
(np.exp(alpha_y + beta_x

return np.exp(-xi*t) * (np.log(rho_correction(x, y, lambda_x, mu_y,
+ ((rho

np.log(poisson.pmf(x, lambda_x)) +
(((np.log(poisson.pmf(y, mu_y

epl_1718 = pd.read_csv("http://www.football-
("data.co.uk/mmz4281/1718/E0.csv

epl_1718['Date'] = pd.to_datetime(epl_1718['Date'],
('format='%d/%m/%y

epl_1718['time_diff'] = (max(epl_1718['Date']) - epl_1718['Date']).dt.days

epl_1718 = epl_1718[['HomeTeam','AwayTeam','FTHG','FTAG', 'FTR',
[['time_diff

epl_1718 = epl_1718.rename(columns={'FTHG': 'HomeGoals', 'FTAG':
({'AwayGoals

()epl_1718.head

time_diff  FTR  AwayGoals  HomeGoals  AwayTeam  HomeTeam
275      H      3      4   Leicester    Arsenal      0
274      A      2      0   Man City    Brighton      1
274      A      3      2    Burnley    Chelsea      2
274      A      3      0   Huddersfield  Crystal Palace      3
274      H      0      1   Stoke      Everton      4

```

```

def solve_parameters_decay(dataset, xi=0.001, debug = False,
, {init_vals=None, options={'disp': True, 'maxiter':100

constraints = [{'type':'eq', 'fun': lambda x: sum(x[:20])-20}] ,
:**kwargs

()) teams = np.sort(dataset['HomeTeam'].unique

check for no weirdness in dataset #

()) away_teams = np.sort(dataset['AwayTeam'].unique

:(if not np.array_equal(teams, away_teams

("raise ValueError("something not right

(n_teams = len(teams

:if init_vals is None

random initialisation of model parameters #

init_vals = np.concatenate((np.random.uniform(0,1,(n_teams)), #
attack strength

np.random.uniform(0,-1,(n_teams)), # defence
strength

np.array([0,1.0]) # rho (score correction),
(gamma (home advantage

((

def dc_log_like_decay(x, y, alpha_x, beta_x, alpha_y, beta_y, rho,
:(gamma, t, xi=xi

lambda_x, mu_y = np.exp(alpha_x + beta_y + gamma),
(np.exp(alpha_y + beta_x

return np.exp(-xi*t) * (np.log(rho_correction(x, y, lambda_x, mu_y,
+ ((rho

np.log(poisson.pmf(x, lambda_x)) +
(((np.log(poisson.pmf(y, mu_y

```

```

:(def estimate_paramters(params
  ([score_coefs = dict(zip(teams, params[:n_teams
  (([defend_coefs = dict(zip(teams, params[n_teams:(2*n_teams
  [:rho, gamma = params[-2
  log_like = [dc_log_like_decay(row.HomeGoals, row.AwayGoals,
  ,[score_coefs[row.HomeTeam], defend_coefs[row.HomeTeam
  score_coefs[row.AwayTeam],
  ,[defend_coefs[row.AwayTeam
  rho, gamma, row.time_diff, xi=xi) for row in
 ]()dataset.itertuples
  (return -sum(log_like
  opt_output = minimize(estimate_paramters, init_vals,
  (options=options, constraints = constraints
  :if debug
  sort of hacky way to investigate the output of the optimisation #
  process
  return opt_output
  :else
  + [return dict(zip(["attack_" + team for team in teams
  + [defence_" + team for team in teams"])
  ,['rho', 'home_adv']
  ((opt_output.x
  (params_xi= solve_parameters_decay(epl_1718, xi=0.0018
  C:\Users\DIRDIRI\AppData\Local\Continuum\anaconda3\lib\site-
  packages\ipykernel_launcher.py:19: RuntimeWarning: divide by zero
  encountered in log

```

C:\Users\DIRDIRI\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launcher.py:19: RuntimeWarning: invalid value encountered in log

(Optimization terminated successfully. (Exit mode 0

Current function value: 832.6598925477288

Iterations: 43

Function evaluations: 1956

Gradient evaluations: 43

params_xi

,attack_Arsenal': 1.4593651460722956'}

,attack_Bournemouth': 0.98551283613765683'

,attack_Brighton': 0.69928781168010956'

,attack_Burnley': 0.70431831190395433'

,attack_Chelsea': 1.2374364621077387'

,attack_Crystal Palace': 1.0097589903242024'

,attack_Everton': 0.94292095612817972'

,attack_Huddersfield': 0.46239735411738703'

,attack_Leicester': 1.1875029236609327'

,attack_Liverpool': 1.5541275187521442'

,attack_Man City': 1.7732021459388028'

,attack_Man United': 1.2929629529047928'

,attack_Newcastle': 0.78053285033641984'

,attack_Southampton': 0.77003553613634246'

,attack_Stoke': 0.70057070734386662'

,attack_Swansea': 0.46822884080600669'

,attack_Tottenham': 1.4286245703681737'
,attack_Watford': 0.88735796685491264'
,attack_West Brom': 0.59796943780667744'
,attack_West Ham': 1.0578866806194034'
,defence_Arsenal': -0.90350688807197488'
,defence_Bournemouth': -0.74359641563485857'
,defence_Brighton': -0.88576784878115034'
,defence_Burnley': -1.1823770286264468'
,defence_Chelsea': -1.1924572572147196'
,defence_Crystal Palace': -0.87595090531915076'
,defence_Everton': -0.82109862955384794'
,defence_Huddersfield': -0.84872681894152968'
,defence_Leicester': -0.73455540299397404'
,defence_Liverpool': -1.2116005980564268'
,defence_Man City': -1.5083062975356949'
,defence_Man United': -1.5143203684750592'
,defence_Newcastle': -1.0661409709636933'
,defence_Southampton': -0.85568382082476147'
,defence_Stoke': -0.6876953305361172'
,defence_Swansea': -0.85747531921329301'
,defence_Tottenham': -1.2568197505130652'
,defence_Watford': -0.72640660682168368'
,defence_West Brom': -0.87337838654160449'
,defence_West Ham': -0.66515172274413292'
,home_adv': 0.30317368503677694'
{rho': -0.13185247769791564'

```
xi_vals = [0.0, 0.0002, 0.0004, 0.0006, 0.0008, 0.001, 0.0012, 0.0014,  
,0.0016, 0.0018
```

```
[0.005 ,0.0045 ,0.004 ,0.0035 ,0.0035 ,0.003 ,0.0025 ,0.002
```

I pulled the scores from files on my computer that had been generated #
seperately

```
[] = xi_scores#
```

```
:for xi in xi_vals#
```

```
:with open ('find_xi_{}.txt'.format(str(xi)[2:]), 'rb') as fp #
```

```
((xi_scores.append(sum(pickle.load(fp) #
```

```
xi_scores = [-125.38424297397718, -125.3994150871104, -  
,125.41582329299528, -125.43330024318175, -125.45167361727589
```

```
- ,125.49165987944551- ,125.47148572476918-  
,125.5588181265923- ,125.53570389317336- ,125.51283291929082
```

```
- ,125.64545123148538- ,125.58171066742123-  
,125.78763678848986- ,125.78763678848986- ,125.71506317675832
```

```
- ,125.94721517841089- ,125.8651515986525-  
[126.03247674382676
```

```
((fig, ax1 = plt.subplots(1, 1, figsize=(10,4
```

```
ax1.plot(xi_vals, xi_scores, label='Component 1', color='#F2055C',  
(marker='o
```

```
([ax1.set_ylim([-126.20, -125.20
```

```
([ax1.set_xlim([-0.0001,0.0051
```

```
([])ax1.set_xticklabels#
```

```
(ax1.set_ylabel('S( $\xi$ )', fontsize=13
(ax1.set_xlabel('xi', fontsize=13
(ax1.xaxis.set_tick_params(labels=12
(ax1.yaxis.set_tick_params(labels=12
ax1.set_title("Predictive Profile Log-Likelihood (EPL 2017/18
('Season)",size=14,fontweight='bold
())plt.show
```

```
()epl_1318 = pd.DataFrame
:(for year in range(13,18
epl_1318 = pd.concat((epl_1318, pd.read_csv("http://www.football-
((((data.co.uk/mmz4281/{}/E0.csv".format(year, year+1
epl_1318['Date'] = pd.to_datetime(epl_1318['Date'],
('format='%d/%m/%y
epl_1318['time_diff'] = (max(epl_1318['Date']) - epl_1318['Date']).dt.days
epl_1318 = epl_1318[['HomeTeam','AwayTeam','FTHG','FTAG', 'FTR',
[['time_diff
epl_1318 = epl_1318.rename(columns={'FTHG': 'HomeGoals', 'FTAG':
({'AwayGoals
('epl_1318 = epl_1318.dropna(how='all
())epl_1318.head
```

time_diff	FTR	AwayGoals	HomeGoals	AwayTeam	HomeTeam
1730.0	A	3.0	1.0	Aston Villa	Arsenal
1730.0	H	0.0	1.0	Stoke	Liverpool
1730.0	D	2.0	2.0	Everton	Norwich
1730.0	A	1.0	0.0	Fulham	Sunderland


```
1730.0      A      4.0      1.0 Man United      Swansea      4
```

```
xi_vals = [0.0, 0.0005, 0.001, 0.0015, 0.002, 0.0025, 0.00275, 0.003,  
,0.00325
```

```
[0.006 ,0.0055 ,0.005 ,0.0045 ,0.00425 ,0.004 ,0.00375 ,0.0035
```

```
I pulled the scores from files on my computer that had been generated #  
seperately
```

```
[] = xi_scores#
```

```
:for xi in xi_vals#
```

```
:with open ('find_xi_5season_{}.txt'.format(str(xi)[2:]), 'rb') as fp      #
```

```
((xi_scores.append(sum(pickle.load(fp)      #
```

```
xi_scores = [-127.64548699733858, -126.88558052909376, -  
,126.24253680407995, -125.75657140537645, -125.43198691100818
```

```
- ,125.1929173322124- ,125.24473381373896-  
,125.15741294807299- ,125.15259048041912- ,125.16314084998176
```

```
- ,125.20427802084305- ,125.17611832471187-  
,125.39161839279092- ,125.2863163741079- ,125.24143128833828
```

```
[125.64269122223465- ,125.51241118364625-
```

```
((fig, ax1 = plt.subplots(1, 1, figsize=(10,4
```

```
ax1.plot(xi_vals, xi_scores, label='Component 1', color='#F2055C',  
(marker='o
```

```
([ax1.set_ylim([-0.05,1.05#
```

```
([ax1.set_xlim([-0.0001, 0.0061
```

```
([])ax1.set_xticklabels#
```

```
(ax1.set_ylabel('S( $\xi$ )', fontsize=13
```

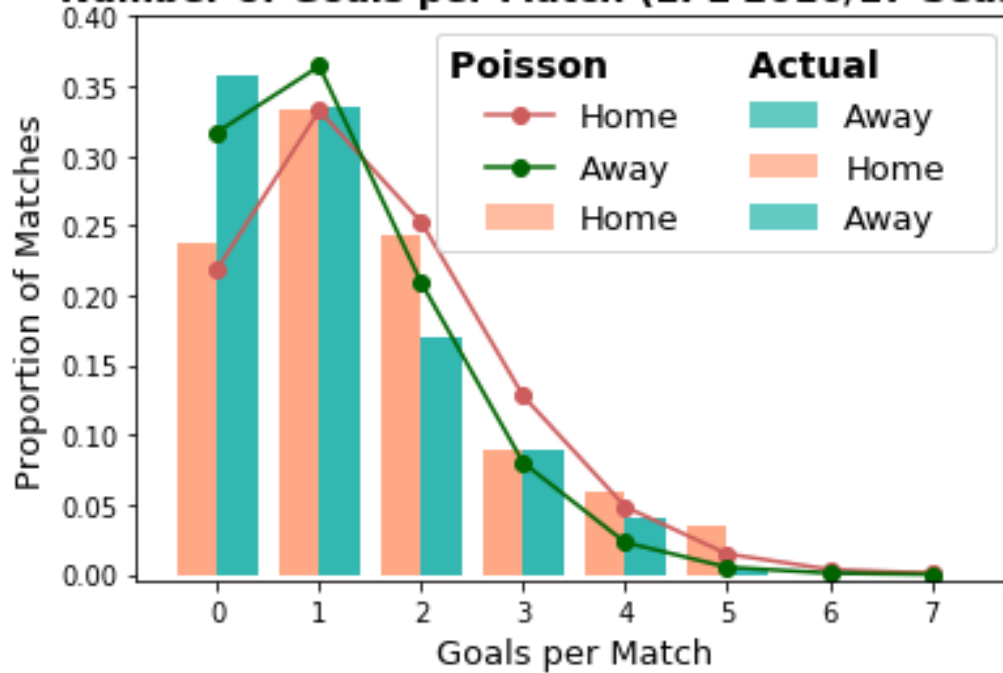
```
(ax1.set_xlabel(' $\xi$ ', fontsize=13
```

```
(ax1.xaxis.set_tick_params(labels=12
```

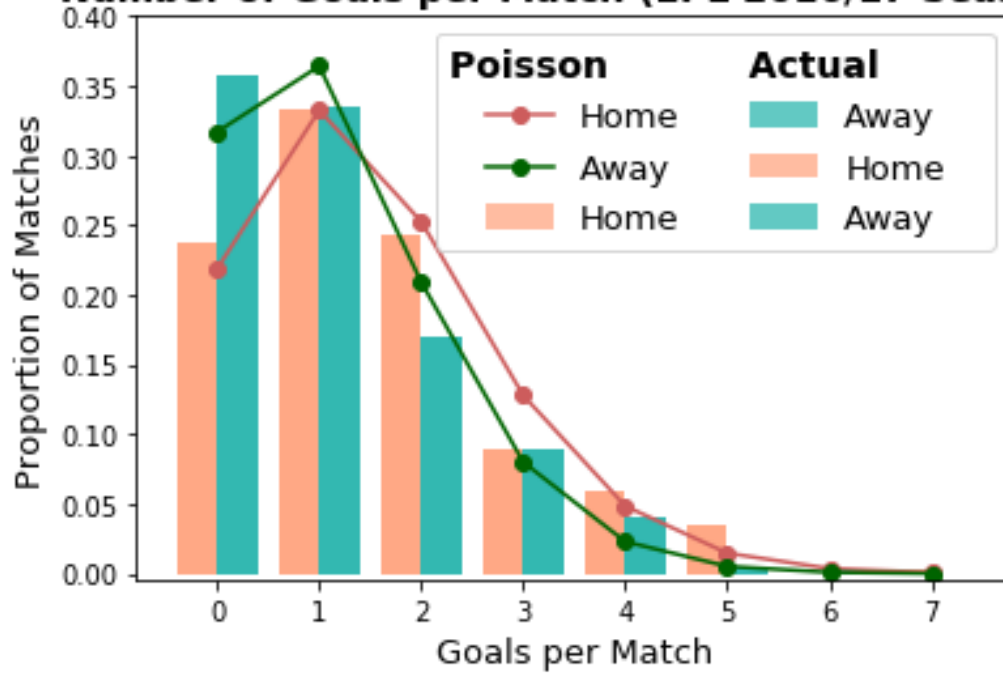
```
(ax1.yaxis.set_tick_params(labels=12
```

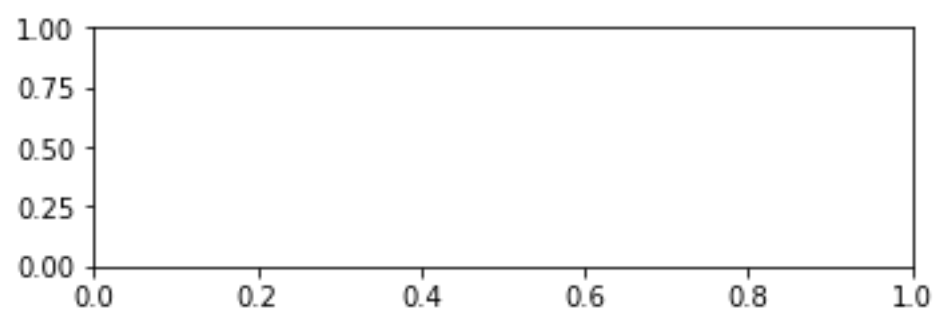
```
ax1.set_title("Predictive Profile Log-Likelihood (EPL 13/14 - 17/18  
(Seasons)",size=14,fontweight='bold
```

Number of Goals per Match (EPL 2016/17 Season)

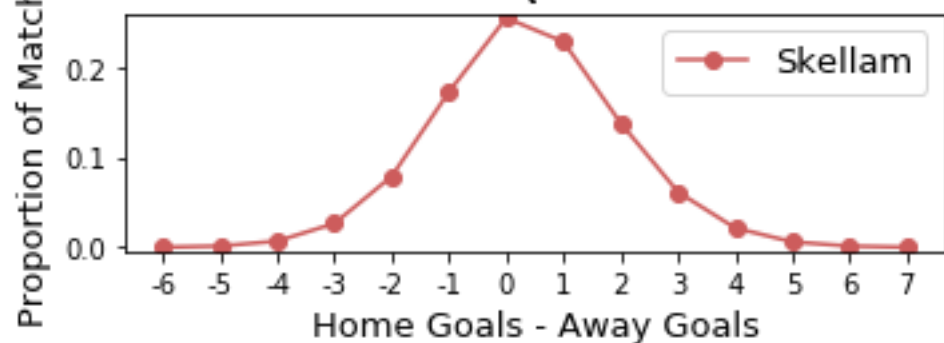


Number of Goals per Match (EPL 2016/17 Season)

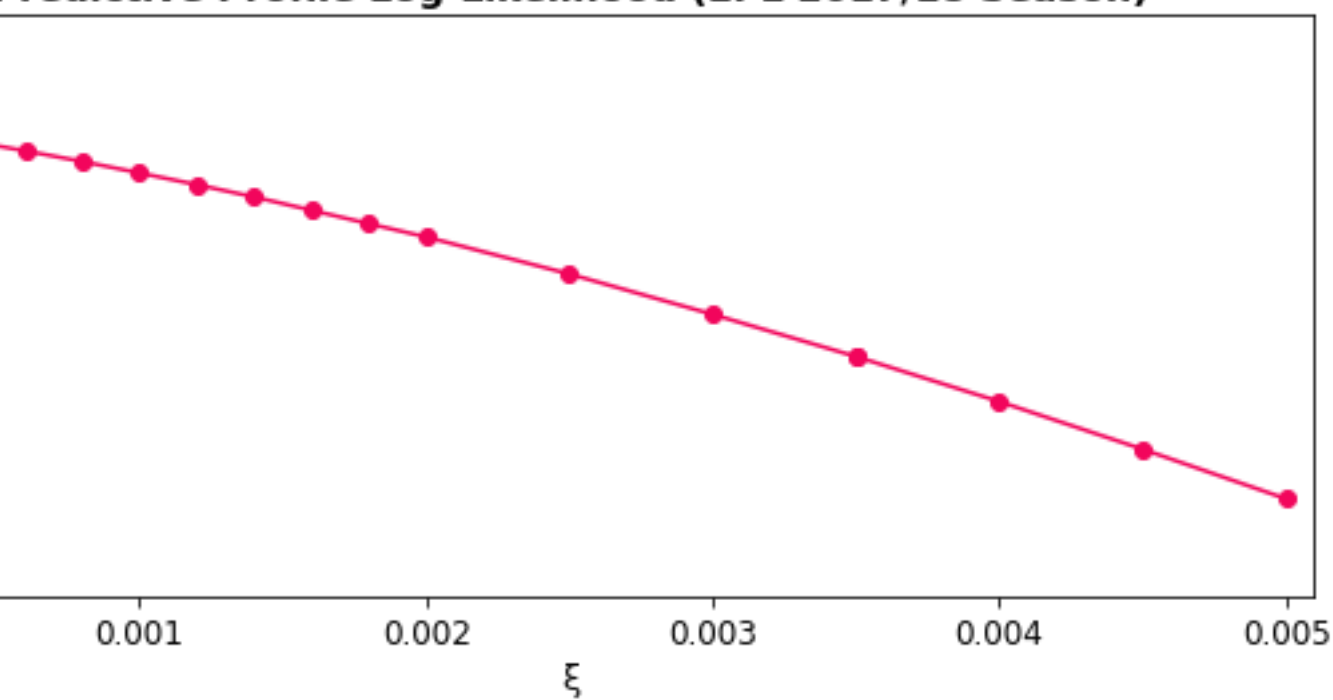




Difference in Goals Scored (Home Team vs Away Team)

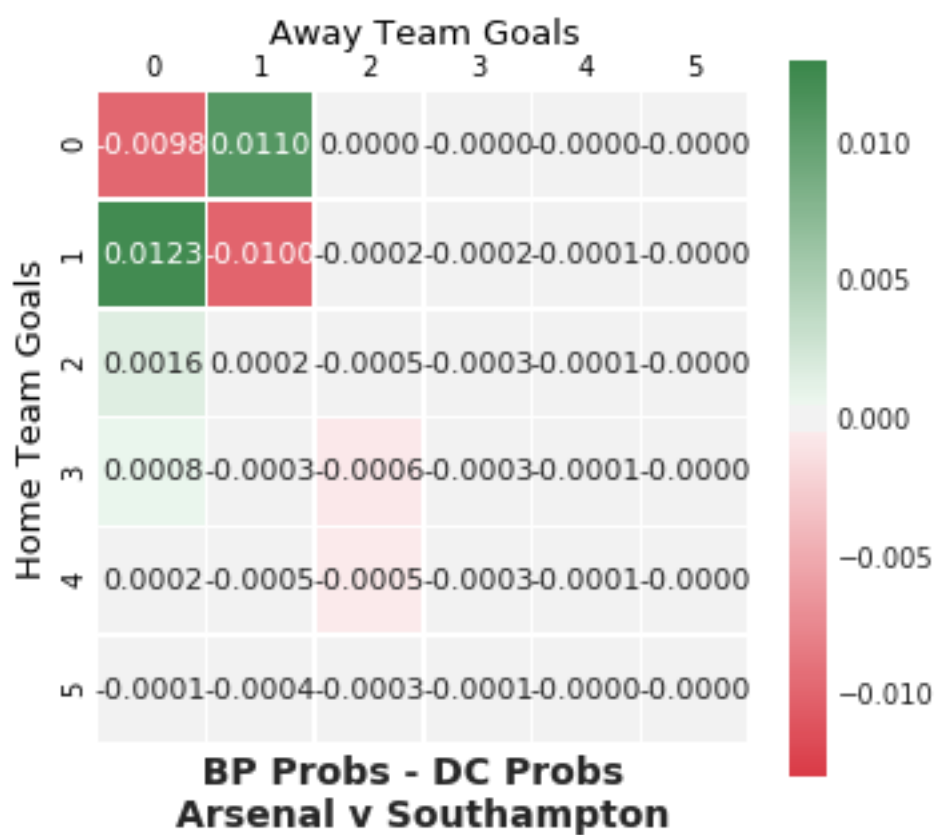
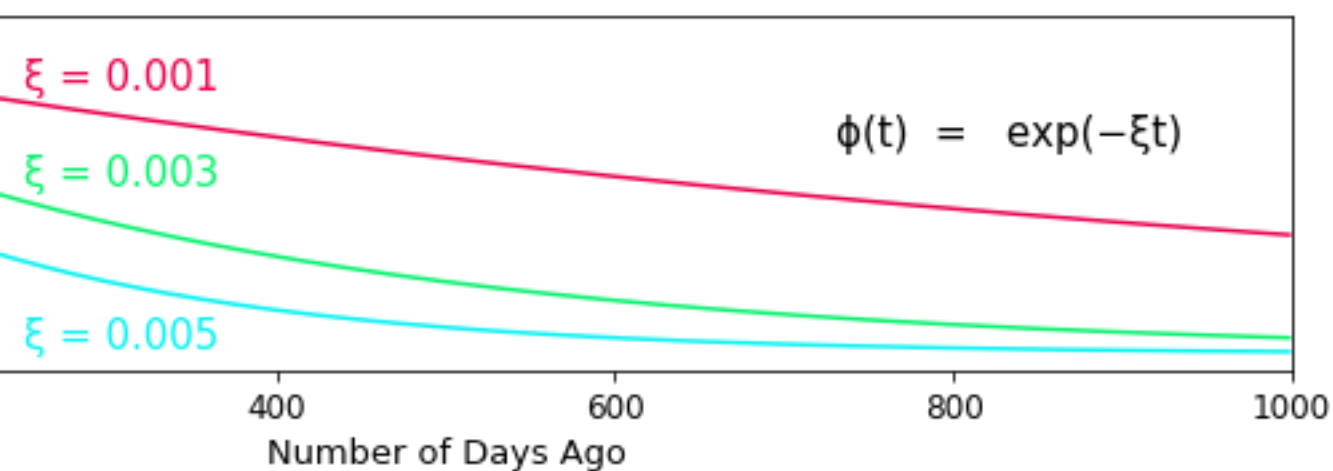


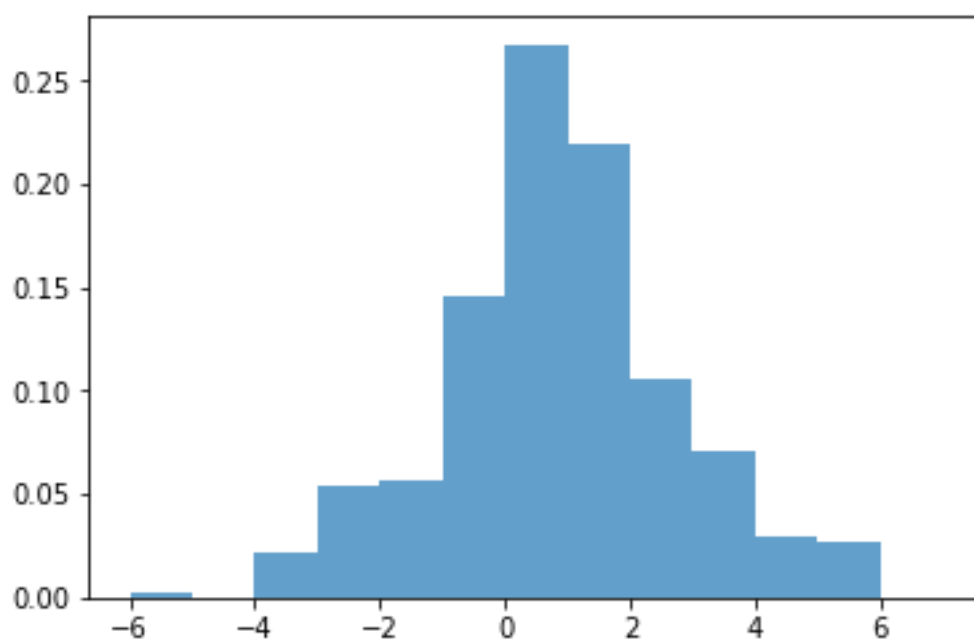
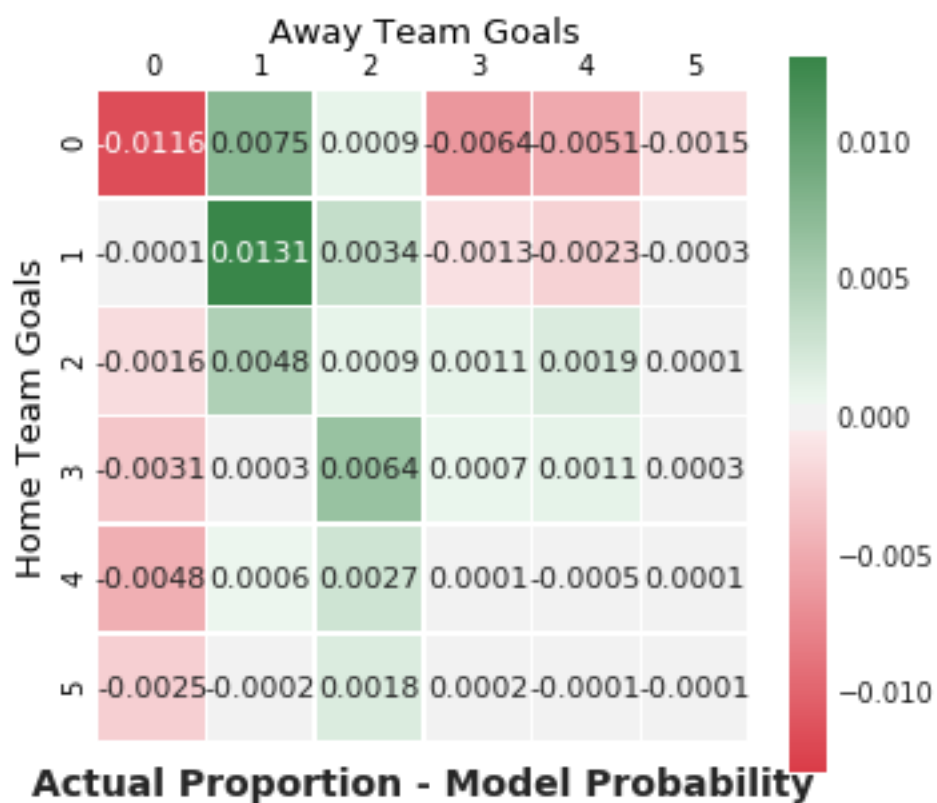
Predictive Profile Log-Likelihood (EPL 2017/18 Season)

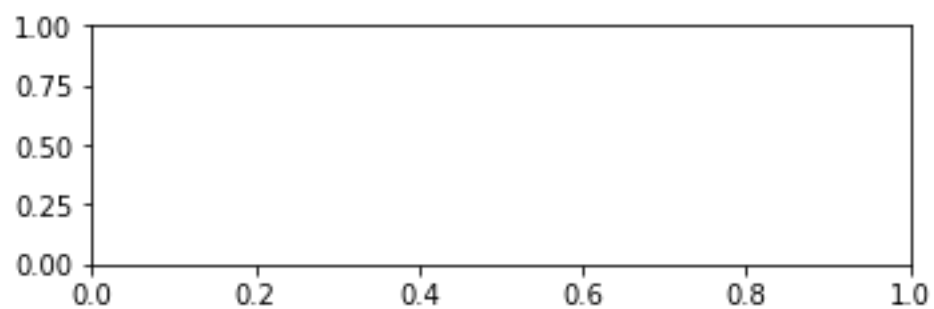


Time Decay Weighting Functions

$$\phi(t) = \begin{cases} 1 & t \leq t_0 \\ 0 & t > t_0 \end{cases}$$







Difference in Goals Scored (Home Team vs Away Team)

