1. Introduction

Betting has always been an exciting and entertaining factor, especially in sports. Over time, this element has formed an increasingly serious industry. Besides, it is based on highly complex modeling and analysis. Afterwards, different prediction bets were created and betting sites offering odds of these types were established.

In this project, we have many different odd data containing different bet types from different bookmakers for English Premier League match results. We are expected to forecast the probabilities for home win, tie and away win of the last season by using these odds, and other possible models. First and foremost, it can be stated that it is a classification problem; however, considering the outputs which can be more than two, it should be called as multi-classification problem. Beforehand applying a model to solve this classification problem, first we used the preprocessing feature data set and determined which columns are required. This data helped us to move on to the selected models more easily. For instance, we converted epoch times into real dates and separated data as train and test (2018-2019 season). After that, we investigated which models can be used to make a better forecasting. First, we tried ***glmnet*** and ***random forest*** by getting support from caret package and calculated by Ranked Probability Score (RPS). In addition, ordinal logistic regression approach was evaluated considering it is a multi-classification problem.

In short, all three models gave roughly comparable results, around 0.30 RPS, and we believe they are beneficial in sports forecasting because of their mixed combination abilities.

2. Literature Summary

An ordered probit regression model estimated using 10 years' data is used to forecast English league football match results. As well as past match results data, the significance of the match for end-of-season league outcomes, the involvement of the teams in cup competition and the geographical distance between the two teams' home towns all contribute to the forecasting model's performance. The model is used to test the weak-form efficiency of prices in the fixed-odds betting market. A strategy of selecting end-of-season bets with a favorable expected return according to the model appears capable of generating a positive return.

The paper can be found in the following link:

- Forecasting Football Results and the Efficiency of Fixed-odds Betting

The efficiency of gambling markets has frequently been questioned. In order to investigate the rationality of bookmaker odds, we use an ordered probit model to generate predictions for English football matches and compare these predictions with the odds of UK bookmaker William Hill. Further, we develop a model that predicts bookmaker odds. Combining a predictive model based on results and a bookmaker model based on previous quoted odds allows us to compare directly William Hill opinion of various teams with the team ratings generated by the predictive model. We also compare the objective value of individual home advantage and distance travelled with the value attributed to these factors by bookmakers. We show that there are systematic biases in bookmaker odds, and that these biases cannot be explained by William Hill odds omitting valuable, or excluding extraneous, information.

The paper can be found in the following link:

- Predicting bookmaker odds and efficiency for UK football

3. Approach

We investigated the possible models which can be efficiently handled for this type of prediction. To understand the performance of the output, we took into account Ranked Probability Score (RPS) of each model.

After getting preprocessed feature set, we considered the case and first decided to apply ***glmnet*** and ***random forest*** approaches, respectively. At this point, we installed caret package in order to make solid tunings while applying these models. The caret package is one of the most popular packages and a comprehensive framework for building machine learning models in R. It includes functions to streamline the model training process for complex regression and classification problems. Apart from these beneficial containments, it can be easily said that the package is very user-friendly.

In addition to that, we used cross validation to make sure almost all possibilities are combined while calculating home, tie, and away probabilities of 2018-2019 English Premier League Matches. This method works with adding a number value, number=5 for instance, the system separates the folds and randomly picks 4 out of 5 the inputs as train data, and test data for the rest. When steadily increased lambda value added, there are many combinations can be evaluated and solicited probabilities are predicted more accurate. For the first two models, "glmnet" and "rf" functions were used and both RPS values were nearly 0.30 for that reason these can be counted as very successful approaches.

Lastly, we wanted to try ***ordinal logistic regression*** model since this case is a multi-classification problem. The function, called as "polr", worked also well. Considering the RPS values of the models, it can be mentioned that ordinal logistic regression (or probit regression) is very close to the other two models, therefore all of them can be thought as efficient approaches.

4. Results

We selected open & close odds of 1x2 games as input to our models based on experiences from previous homeworks as bookmaker odds already takes all team conditions and past data into account. Considering our problem is an ordinal regression model, we selected 3 different methods: glmnet, random forest and Ordered Logistic Regression where average RPS values are 0.3276606, 0.3192165, 0.3215795 respectively. The model with lowest RPS value hence the best prediction among the tried models is the Random Forest. Below are the results of the test data with Random Forest Model.

> results

	P(Away)	P(Home)	P(Tie)	Match_Result	RPS
1	0.014	0.946	0.040	Home	0.486898
2	0.102	0.618	0.280	Away	0.264402
3	0.274	0.474	0.252	Home	0.295290
4	0.188	0.584	0.228	Tie	0.043664
5	0.112	0.698	0.190	Home	0.412322

6	0.184	0.618	0.198	Tie	0.036530
7	0.176	0.656	0.168	Away	0.361600
8	0.290	0.400	0.310	Home	0.300100
9	0.252	0.350	0.398	Away	0.212954
10	0.226	0.438	0.336	Home	0.355986
11	0.000	0.952	0.048	Home	0.501152
12	0.236	0.532	0.232	Home	0.318760
13	0.502	0.284	0.214	Tie	0.148900
14	0.386	0.348	0.266	Home	0.223876
15	0.464	0.212	0.324	Away	0.336136
16	0.238	0.502	0.260	Home	0.324122
17	0.302	0.470	0.228	Home	0.269594
18	0.172	0.638	0.190	Away	0.342842
19	0.264	0.460	0.276	Away	0.296936
20	0.116	0.774	0.110	Tie	0.012778
21	0.318	0.386	0.296	Away	0.298370
22	0.318	0.268	0.414	Away	0.222260
23	0.580	0.214	0.206	Home	0.109418
24	0.458	0.344	0.198	Away	0.426484
25	0.648	0.190	0.162	Away	0.561074
26	0.330	0.412	0.258	Away	0.329732
27	0.244	0.514	0.242	Home	0.315050
28	0.270	0.430	0.300	Home	0.311450
29	0.304	0.414	0.282	Home	0.281970
30	0.288	0.598	0.114	Home	0.259970
31	0.060	0.730	0.210	Home	0.463850
32	0.002	0.814	0.184	Home	0.514930
33	0.252	0.336	0.412	Tie	0.116624
34	0.234	0.652	0.114	Home	0.299876
35	0.150	0.758	0.092	Home	0.365482
36	0.194	0.524	0.282	Away	0.276580
37	0.698	0.086	0.216	Away	0.550930
38	0.298	0.392	0.310	Away	0.282452
39	0.182	0.598	0.220	Tie	0.040762
40	0.242	0.478	0.280	Home	0.326482
41	0.230	0.424	0.346	Away	0.240308
42	0.108	0.822	0.070	Home	0.400282
43	0.296	0.444	0.260	Away	0.317608
44	0.174	0.708	0.118	Home	0.348100
45	0.314	0.304	0.382	Home	0.308260

46	0.130	0.620	0.250	Home	0.409700
47	0.296	0.412	0.292	Tie	0.086440
48	0.222	0.654	0.124	Tie	0.032330
49	0.150	0.708	0.142	Home	0.371332
50	0.052	0.694	0.254	Tie	0.033610
51	0.114	0.810	0.076	Home	0.395386
52	0.264	0.266	0.470	Home	0.381298
53	0.068	0.746	0.186	Home	0.451610
54	0.246	0.310	0.444	Tie	0.128826
55	0.176	0.596	0.228	Home	0.365480
56	0.206	0.528	0.266	Tie	0.056596
57	0.150	0.744	0.106	Home	0.366868
58	0.258	0.426	0.316	Tie	0.083210
59	0.190	0.396	0.414	Away	0.189748
60	0.170	0.582	0.248	Tie	0.045202
61	0.288	0.458	0.254	Tie	0.073730
62	0.566	0.276	0.158	Away	0.514660
63	0.882	0.024	0.094	Away	0.799380
64	0.162	0.356	0.482	Away	0.147284
65	0.626	0.140	0.234	Away	0.489316
66	0.276	0.338	0.386	Away	0.226586
67	0.142	0.726	0.132	Tie	0.018794
68	0.216	0.392	0.392	Home	0.384160
69	0.442	0.330	0.228	Home	0.181674
70	0.412	0.310	0.278	Away	0.345514
71	0.002	0.976	0.022	Home	0.498244
72	0.314	0.416	0.270	Away	0.315748
73	0.458	0.168	0.374	Home	0.216820
74	0.030	0.840	0.130	Home	0.478900
75	0.234	0.506	0.260	Away	0.301178
76	0.060	0.826	0.114	Away	0.394298
77	0.358	0.450	0.192	Home	0.224514
78	0.496	0.180	0.324	Away	0.351496
79	0.112	0.466	0.422	Away	0.173314
80	0.626	0.212	0.162	Away	0.547060
81	0.612	0.166	0.222	Away	0.489914
82	0.252	0.316	0.432	Away	0.193064
83	0.182	0.656	0.162	Home	0.347684
84	0.860	0.028	0.112	Away	0.764072
85	0.102	0.818	0.080	Home	0.406402

86	0.302	0.410	0.288	Away	0.299074
87	0.644	0.188	0.168	Away	0.553480
88	0.172	0.582	0.246	Tie	0.045050
89	0.082	0.786	0.132	Home	0.430074
90	0.334	0.406	0.260	Home	0.255578
91	0.244	0.232	0.524	Tie	0.167056
92	0.000	0.898	0.102	Home	0.505202
93	0.228	0.526	0.246	Tie	0.056250
94	0.760	0.108	0.132	Away	0.665512
95	0.144	0.602	0.254	Home	0.398626
96	0.360	0.316	0.324	Home	0.257288
97	0.232	0.448	0.320	Away	0.258112
98	0.164	0.760	0.076	Home	0.352336
99	0.372	0.274	0.354	Tie	0.131850
100	0.400	0.222	0.378	Tie	0.151442
101	0.302	0.318	0.380	Home	0.315802
102	0.552	0.206	0.242	Home	0.129634
103	0.148	0.514	0.338	Home	0.420074
104	0.306	0.446	0.248	Away	0.329570
105	0.260	0.444	0.296	Away	0.281608
106	0.030	0.850	0.120	Tie	0.007650
107	0.400	0.356	0.244	Away	0.365768
108	0.296	0.244	0.460	Home	0.353608
109	0.160	0.592	0.248	Home	0.383552
110	0.010	0.544	0.446	Home	0.589508
111	0.546	0.238	0.216	Away	0.456386
112	0.424	0.256	0.320	Away	0.321088
113	0.630	0.200	0.170	Away	0.542900
114	0.088	0.734	0.178	Home	0.431714
115	0.360	0.268	0.372	Tie	0.133992
116	0.394	0.368	0.238	Away	0.367940
117	0.022	0.856	0.122	Home	0.485684
118	0.552	0.166	0.282	Tie	0.192114
119	0.298	0.416	0.286	Tie	0.085300
120	0.632	0.192	0.176	Away	0.539200
121	0.086	0.776	0.138	Home	0.427220
122	0.190	0.686	0.124	Home	0.335738
123	0.192	0.550	0.258	Home	0.359714
124	0.732	0.140	0.128	Away	0.648104
125	0.330	0.290	0.380	Away	0.246650

126	0.304	0.364	0.332	Home	0.297320
127	0.770	0.048	0.182	Away	0.631012
128	0.294	0.416	0.290	Away	0.295268
129	0.354	0.388	0.258	Tie	0.095940
130	0.170	0.732	0.098	Home	0.349252
131	0.376	0.304	0.320	Home	0.245888
132	0.274	0.298	0.428	Away	0.201130
133	0.214	0.632	0.154	Home	0.320756
134	0.334	0.472	0.194	Home	0.240596
135	0.196	0.596	0.208	Home	0.344840
136	0.088	0.824	0.088	Home	0.419744
137	0.626	0.180	0.194	Away	0.520756
138	0.412	0.374	0.214	Tie	0.107770
139	0.230	0.424	0.346	Tie	0.086308
140	0.196	0.344	0.460	Tie	0.125008
141	0.598	0.204	0.198	Tie	0.198404
142	0.606	0.142	0.252	Home	0.109370
143	0.754	0.086	0.160	Away	0.637058
144	0.416	0.248	0.336	Away	0.306976
145	0.562	0.156	0.282	Away	0.415684
146	0.280	0.486	0.234	Tie	0.066578
147	0.552	0.148	0.300	Away	0.397352
148	0.206	0.554	0.240	Home	0.344018
149	0.656	0.136	0.208	Away	0.528800
150	0.348	0.366	0.286	Tie	0.101450
151	0.084	0.692	0.224	Home	0.444616
152	0.012	0.728	0.260	Home	0.521872
153	0.292	0.282	0.426	Away	0.207370
154	0.016	0.902	0.082	Away	0.421490
155	0.652	0.192	0.156	Tie	0.224720
156	0.736	0.100	0.164	Home	0.048296
157	0.252	0.544	0.204	Away	0.348560
158	0.920	0.006	0.074	Away	0.851938
159	0.312	0.282	0.406	Away	0.225090
160	0.694	0.114	0.192	Away	0.567250
161	0.082	0.772	0.146	Home	0.432020
162	0.246	0.466	0.288	Home	0.325730
163	0.252	0.492	0.256	Tie	0.064520
164	0.246	0.506	0.248	Home	0.315010
165	0.582	0.308	0.110	Away	0.565412

166	0.396	0.392	0.212	Home	0.204880
167	0.168	0.562	0.270	Home	0.382562
168	0.246	0.440	0.314	Tie	0.079556
169	0.342	0.278	0.380	Home	0.288682
170	0.724	0.154	0.122	Away	0.647530
171	0.098	0.528	0.374	Tie	0.074740
172	0.208	0.494	0.298	Home	0.358034
173	0.036	0.910	0.054	Home	0.466106
174	0.174	0.604	0.222	Home	0.365780
175	0.278	0.370	0.352	Home	0.322594
176	0.592	0.248	0.160	Away	0.528032
177	0.006	0.946	0.048	Home	0.495170
178	0.094	0.810	0.096	Home	0.415026
179	0.366	0.238	0.396	Tie	0.145386
180	0.748	0.126	0.126	Away	0.661690

```
> mean(RPS)
[1] 0.3192165
```

5.Conclusion and future work

To conclude, Random Forest was the best model with lowest RPS compared to glmnet and Ordered Logistic Regression. In our model, we took 1x2 odds of different bookmakers as the feature set since bookmakers already take all possibilities into account professionally while calculating odds for different bets.

Some improvements may be done if the variances between different bookmakers and odds of different types of bets are also taken into account, but we do not expect much difference. Also, data from some bookmakers can be filtered out selecting the bookmakers with best guesses.

Finally, combinations of different methods can be tried to see if it would give any improvement.

5.Code

All the code can be found here:

https://github.com/pjournal/etm01-aslierdal/tree/master/Project