欧洲足球数据分析

该项目代码见 https://github.com/rucnyz/soccer ，代码按照下述说明完成，描述分析主要在 visualization.py 中，建模内容第一次运行（未抽取特征并保存）需要 92 分钟左右。更详细运行介绍见项目的readme。

1 数据集

欧洲足球数据集有着丰富的球队、球员以及比赛数据。具体来说，包括从 2008 年到 2016 年超过 25000 次比赛数据、超过 10000 名球员的数据来自 11 个欧洲国家自己的联赛，其中球员和球队的能力数据来源于 EA 游戏 FIFA 的内容。而且每场比赛还包括 10 个博彩网站的赔率数据。

2 初步思路

2.1 数据集分析

面对非常丰富的数据以及各种各样可选择的任务，首先对数据集进行梳理并明确目标。

总的来说，我们认为和本课程关联最大且最有意义的研究内容包括以下几个部分：

描述分析		**预测任务**
描述球员能力	**描述球队水平**	对比赛结果进行预测（多分类任务）
明星球员和普通球员的区别	主客场胜率分析	
球场不同位置球员的区别	球队水平对比	
不同联赛球员的区别	制作球队排行榜	

原因：该数据集包含大量 FIFA 球员数据以及丰富的球员特征，应当能够得到良好的效果；同时包含 2008-2016 年 25797 条比赛记录，进行分类任务的数据量足够。

而数据集的核心内容可以分为三个部分（*表格待美化*）：

球员能力	**overall_rating** weight height age potential preferred_foot attacking_work_rate defensive_work_rate crossing finishing heading_accuracy short_passing volleys dribbling curve free_kick_accuracy long_passing ball_control acceleration sprint_speed agility reactions balance shot_power jumping stamina strength long_shots aggression interceptions positioning vision penalties marking standing_tackle sliding_tackle gk_diving gk_handling gk_kicking gk_positioning gk_reflexes
球队情况	(buildUpPlaySpeed buildUpPlaySpeedClass buildUpPlayDribbling buildUpPlayDribblingClass buildUpPlayPassing buildUpPlayPassingClass buildUpPlayPositioningClass) (chanceCreationPassing chanceCreationPassingClass chanceCreationCrossing chanceCreationCrossingClass chanceCreationShooting chanceCreationShootingClass chanceCreationPositioningClass) (defencePressure defencePressureClass defenceAggression defenceAggressionClass defenceTeamWidth defenceTeamWidthClass defenceDefenderLineClass)
比赛数据	(**home_player_1~11 away_player_1~11**) (**home_team_goal away_team_goal goal**) (shoton shotoff foulcommit card cross corner possession) (**B365 BW IW ...**)

那么确定了可行任务以及可利用的数据后，接下来就是对该问题进行建模了

2.2 初步研究

分为描述分析和统计建模两个部分。

2.2.1 描述性分析

2.2.1.1 球员能力分析：

球员能力的变量总共包含 42 个，进行描述分析过于复杂，我们决定首先参照 FIFA官网 的选择划分球员的六维能力，即根据球员的过往比赛数据，按照以下方式划分六维属性。

Attacking	Skill	Movement	Power	Mentality	Defending
Crossing	Dribbling	Acceleration	Shot Power	Aggression	Standing Tackle
Finishing	Curve	Sprint Speed	Jumping	Interceptions	Sliding Tackle
Heading Accuracy	Free kick Accuracy	Agility	Stamina	Positioning	
Short Passing	Long Passing	Reactions	Strength	Vision	
Volleys	Ball Control	Balance	Long Shots	Penalties	

同时对同一球员所有比赛的对应维度得分取平均值，将各维度得分映射至 0-10 区间（取该维度最高分为 10)，得到球员最终的各维度得分。

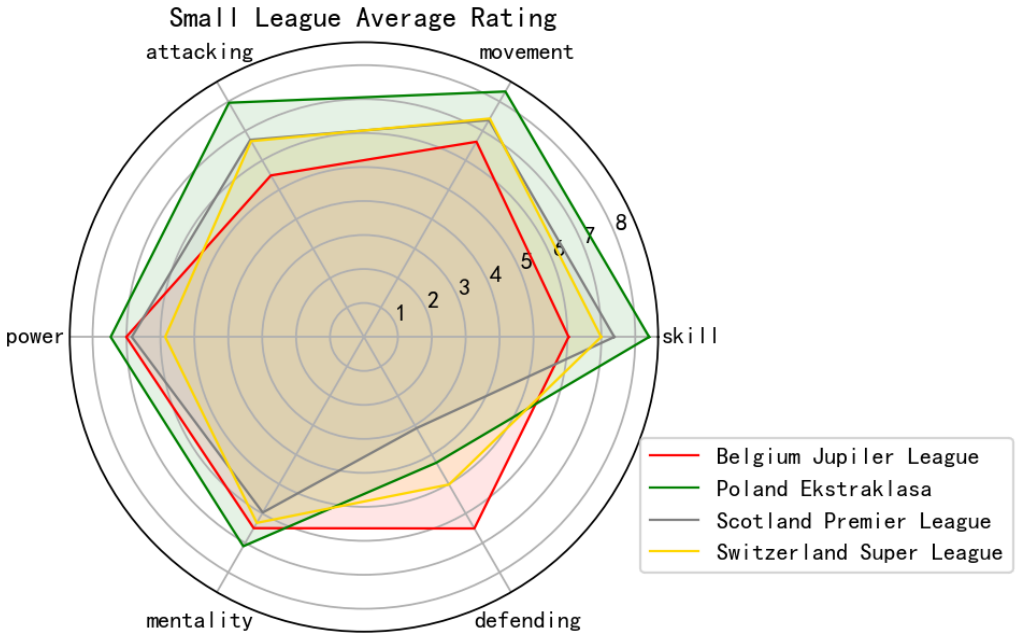
- **不同联赛球员的对比**

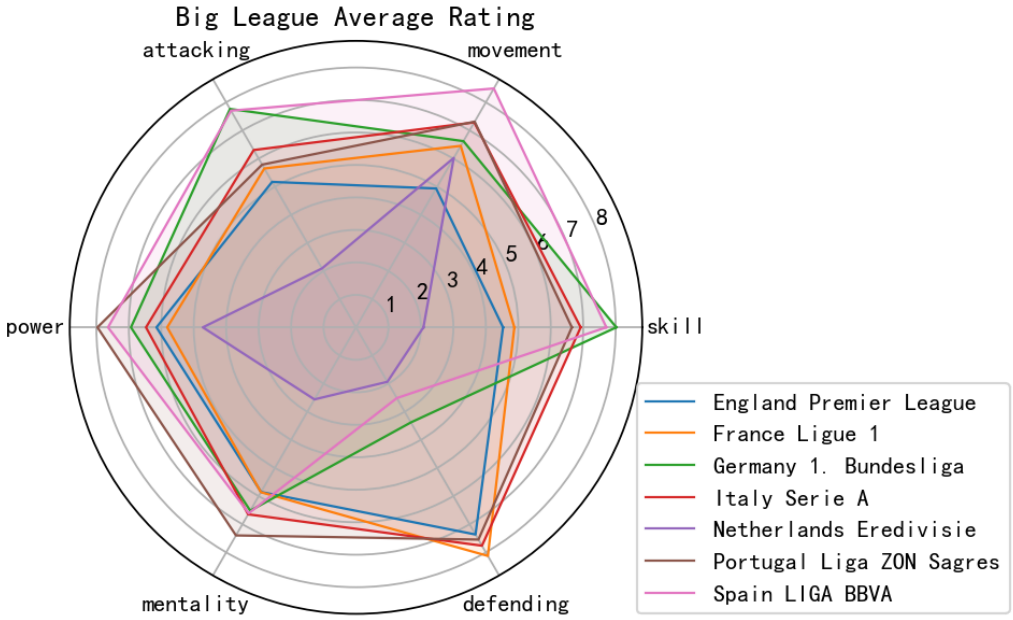
我们希望能够了解不同联赛球员的情况，也即得到不同联赛的水平和各项能力情况。然而联赛共有 11 个，我们因此根据 2008-2016 年进行的比赛场数对各联赛的规模进行了分析，结果如下图所示，红色圆圈越大代表该联赛规模越大。

于是我们依此分开进行分析，即将 11 家欧洲联赛分为两部分，大体量与小体量的联赛。

- Portugal、Spain、England、Germany、Italy、Netherlands、France
- Scotland、Belgium、Switzerland、Poland

依此得到了联赛平均评分如下图所示。





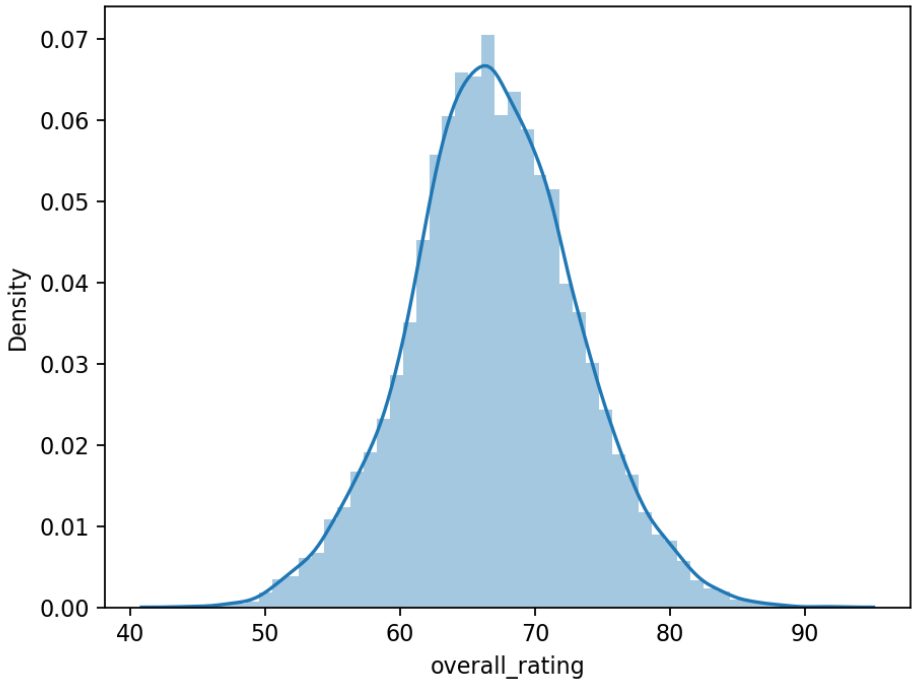
从大联赛的图中可以看出来，荷兰甲级联赛的水平相比于其他联赛最低，只有移动能力上接近其他联赛（*合理怀疑这是罗本一个人带上去的*）。

各项对比情况来看，防守水平上各大联赛出现了较大差别，西甲和德甲的防守水平较低，法甲的防守水平最高。其余各项的各联赛水平较为接近。

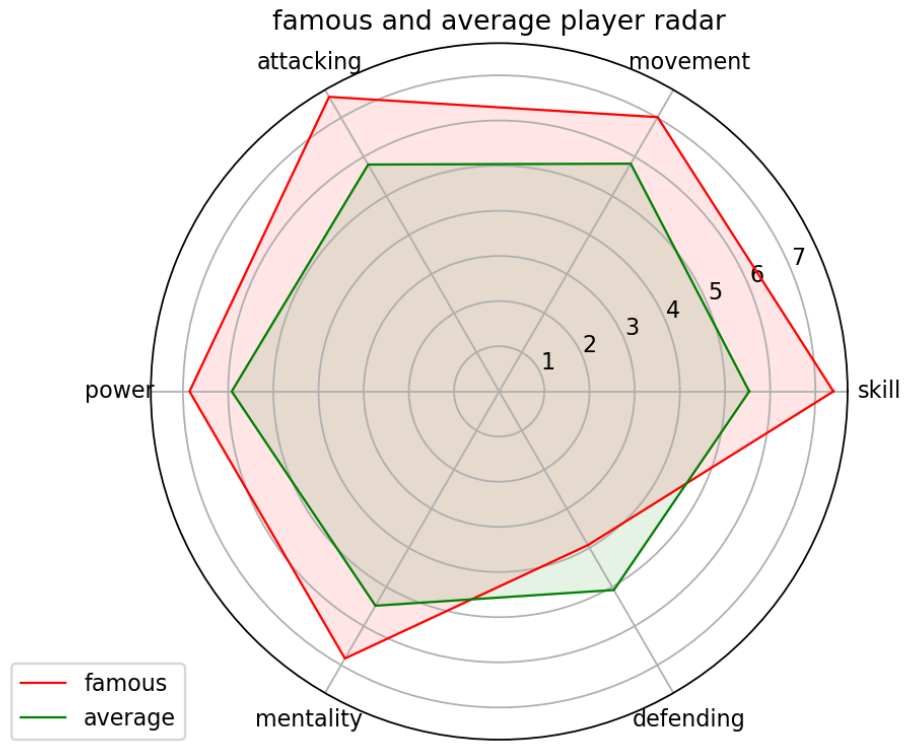
更重要的是，这说明不同联赛的球员水平存在明显的区别，我们将在后续的统计建模过程中把球员隶属的联赛纳入考虑。

- **超级明星球员和普通球员的对比**

为了能够选取尽可能厉害和普通的球员，我们对球员的总体评分绘制了密度图，从图中可以注意到 90 分以上球员数量极少，符合明星球员的定义。而 70 分以下大约占到 60%，应当是普通球员的水平。



我们按照上述分析挑选了总体评分在 90 分以上的球员共 **24** 位（其中包括 C 罗、梅西、罗本、卡卡等在内的顶级球星），以及评分在 70 分以下的球员 **9585** 位（占到总球员数）作为对比，并且选取了他们各自的巅峰期的数据绘制六维图，结果如图所示。



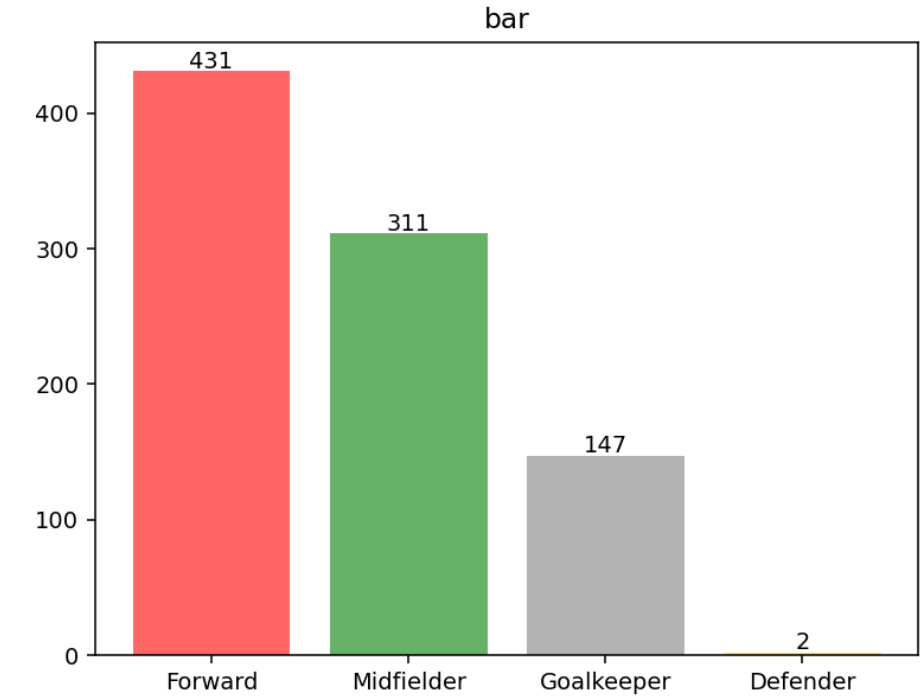
从图中可以发现，除开防守以外的其余所有属性，明星球员都在普通球员之上，而防守能力明星球员却低于普通球员，我们猜测是因为总评分大于 90 分的球员中后卫数量较少，我们在下面的分析中验证了这一点。

- **球场不同位置球员的对比**

在开始这一部分的时候我们遇到了很大的困难，因为数据中只提供了球员每场比赛在球场的二维坐标信息，想要用这个完全分出球场上的所有位置是不现实的，因此在权衡之后我们考虑将球员分为四种类型，按照 Y 轴坐标进行划分：

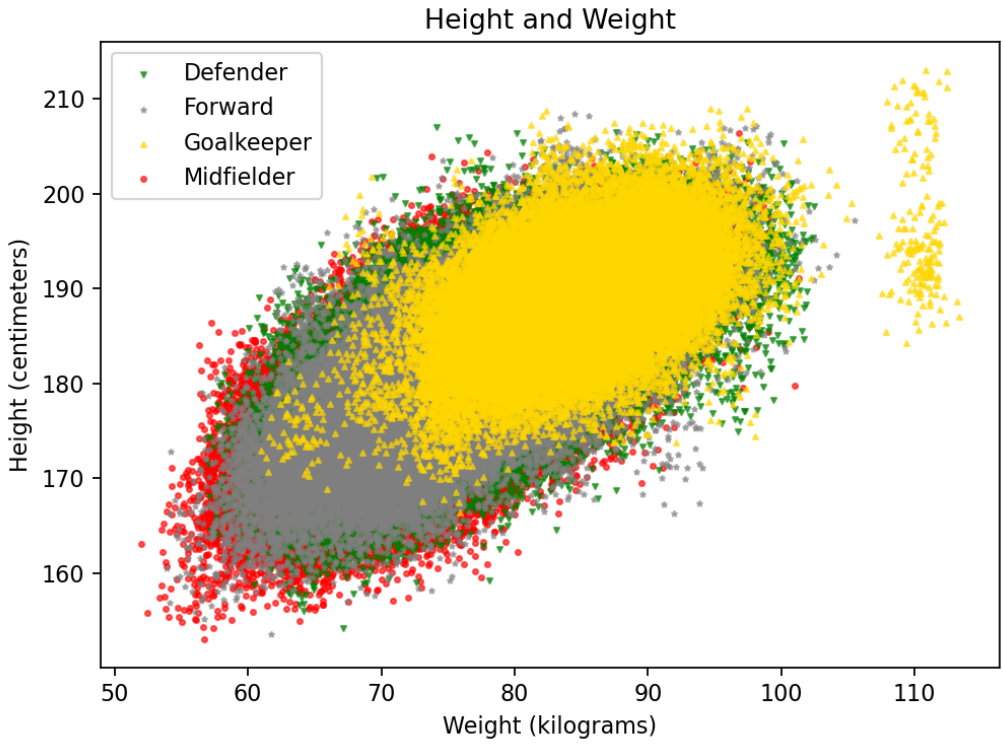
- forward(影\中\边锋等前锋位置)，简称 for。 $Y \geq 10$
- Midfielder(边\前\后腰，边\中\前卫等中场位置)，简称 mid。 $5 < Y < 10$
- Defender(中后卫、边后卫)，简称 def。 $1 < Y \leq 5$
- Goalkeeper(守门员)，简称 gk。 $Y = 1$

首先我们验证了 90 分以上的球员位置分布，我们将所有比赛中任何位置的球员评分大于等于 90 的取出来，一共得到 891 条，其中

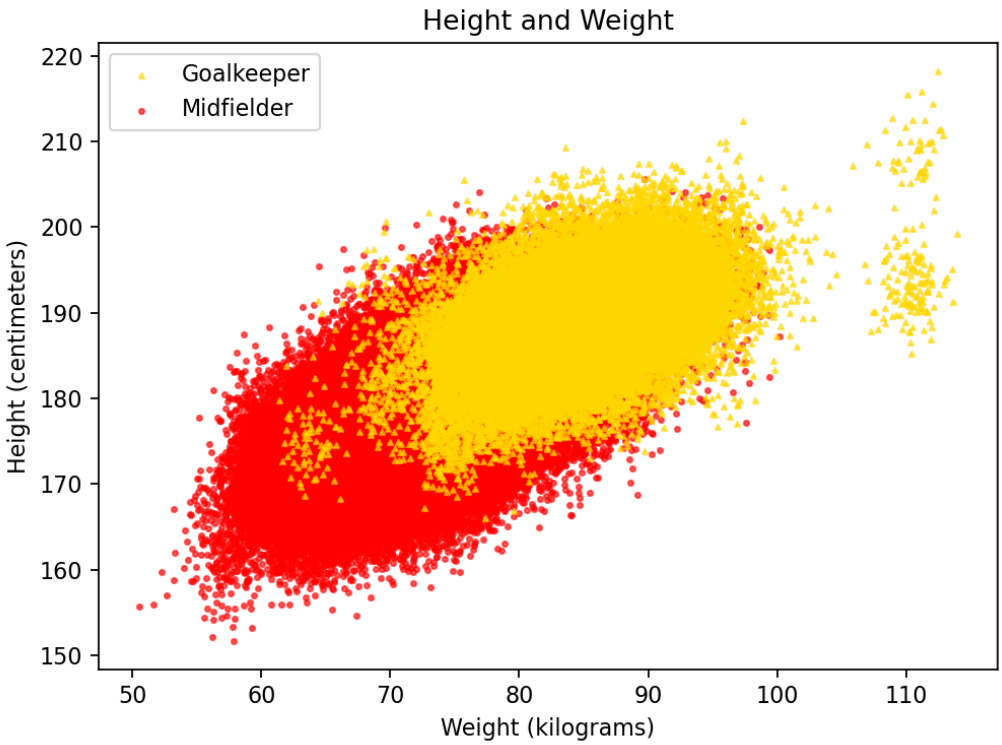


如我们所想，后卫获得 90 以上的评分实在太少了。

接下来我们探究了不同位置球员的身高体重情况。



从身高体重来看，守门员整体偏高偏壮，而中场球员则是整体最偏矮偏轻的，但这张图遮挡比较严重，我们单独挑出中场和守门员可以看的更清楚。

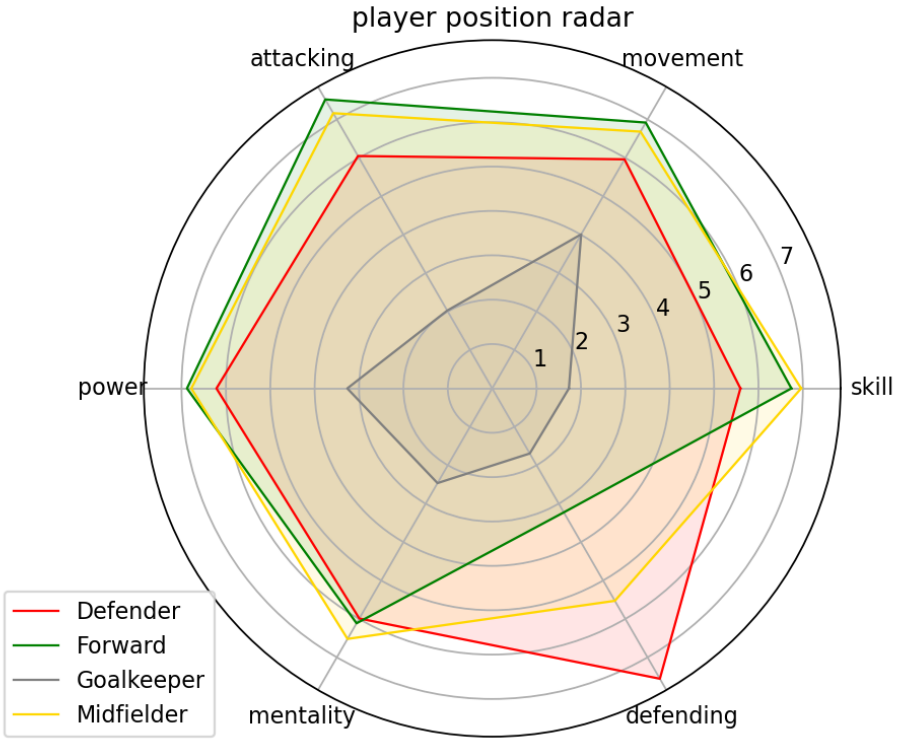


我们认为这和中场球员需要经常且间断性的急停、冲刺、追球有关，他们不像守门员需要足够的身高来保护球门，也不像前锋和后卫频繁的身体对抗需要一定的体重，中场球员对耐力和速度的要求很高，因此可能体重偏轻更加具有优势。

而接下来我们研究了球员的年龄和评分的关系
可以发现一个非常奇怪的现象，居然随着年龄的增加，评分一直在不断上涨。

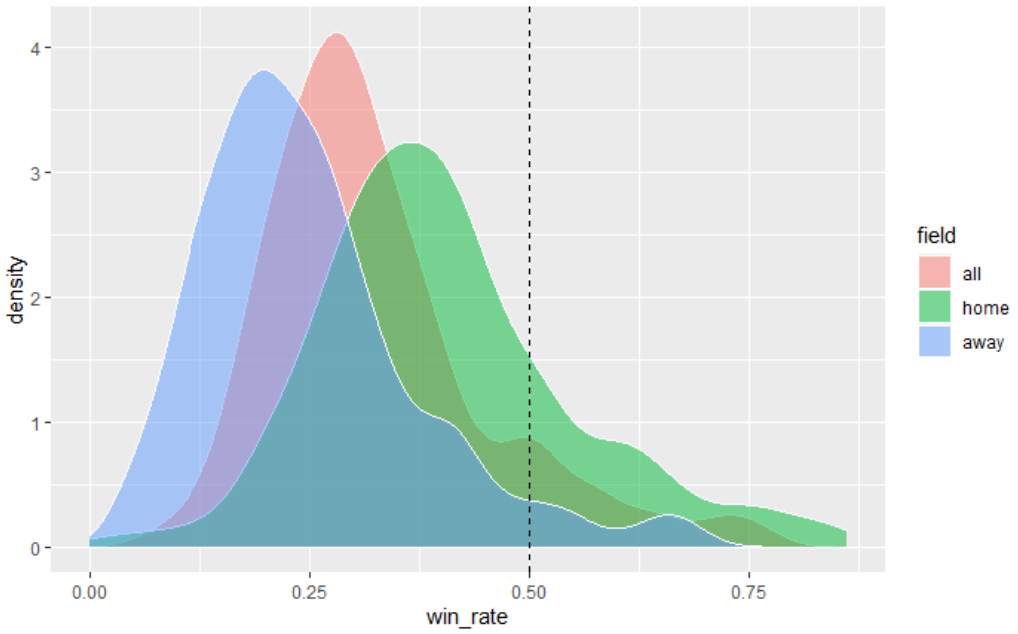
分析之后认为，首先注意到各年龄段平均后的评分都不高，最多也只到了 76 左右，也就是说此时普通球员的评分是左右得分的重要因素，因此我们可以基本得出结论，平均水平下的老将相比于年轻球员具有更强的稳定性。
而且 我们后来注意到，FIFA 的评分系统倾向于给那些成名已久的球员更高的评分，他们相比于年轻球员有着更多的比赛记录，总体上来说获得的评分更高一些。

由于这样的偏差存在，我们在后续的统计建模过程中决定暂时不引入身高体重和年龄因素。



2.2.1.2 球队胜率分析：

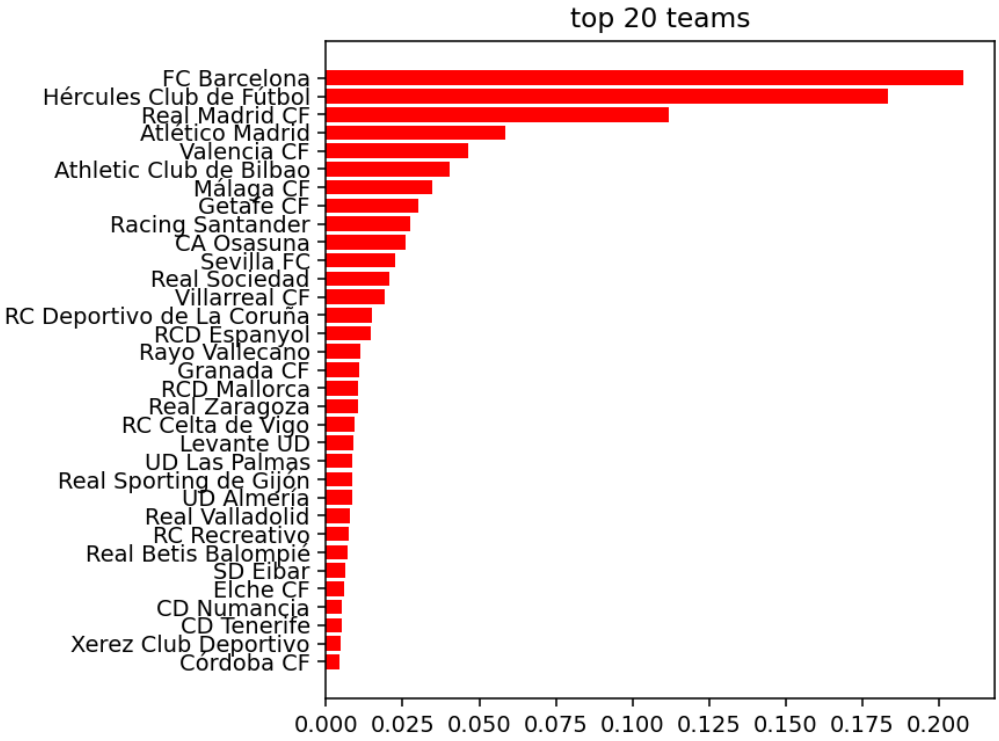
在球队层面，最重要的自然就是胜率。我们首先计算了各个球队胜场及负场数，并对其胜率进行分析，发现只有绝大部分球队胜率低于 40%，只有约 10%的球队胜率大于 50%，这是显而易见的，足球联赛中的二八效应非常明显，豪门实力比普通球队实力强很多。且球队主场胜率明显高于客场胜率，这显著的说明了主客场对胜负情况带来的影响，我们也纳入了后续统计建模的考虑中。



而接下来在针对球队的排名分析，我们考虑了两个方面，一是考虑一个联赛中的所有球队（我们以西甲为例），二则是把各联赛所有球队全部纳入考虑。

- 西甲球队排行

首先我们把球队的胜负关系建成有向图，考虑2008-2016年全部比赛的净胜场数，由败方指向胜方，若净胜场数为0则加上两条边，接着采用pagerank算法得到排行榜如下。



排行榜大体是正确的，然而我们却惊奇的在第二的位置上发现了埃尔库莱斯，这是一只常年混迹于西乙，在2010年进入西甲并打了一年即降级回到西乙的球队。它怎么可能出现在第二的位置上呢？

带着这样的疑问，我们查看了埃尔库莱斯，编号为10278这只球队对其他球队的胜场情况，结果发现这只球队只打过其他12只球队、对8只净胜场为负、4只为正，pagerank算法怎么会把这样的球队排到第4呢？

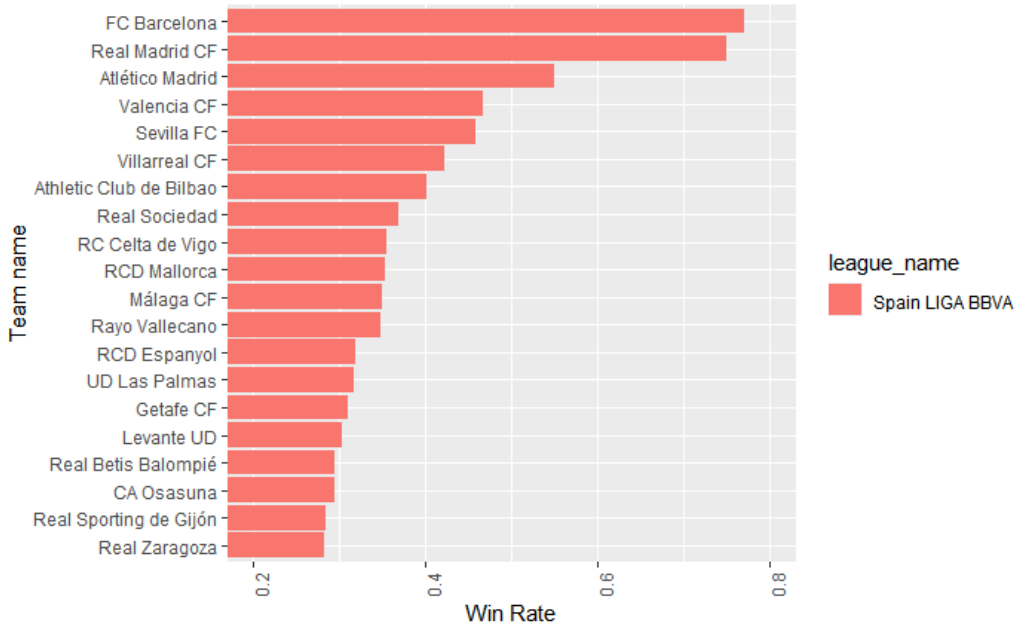
team_1	team_2	outcome
9783	10278	1
8634	10278	1
8302	10278	1
8560	10278	2

我们发现，这只球队仅有的四个净胜场球队中，居然包括了巴塞罗那，即排名第一的球队，由此疑问得以解开。

我们知道PageRank算法本质上是在计算节点被一个随机游走的用户访问的概率，那么即使一只球队输了很多弱队，他一旦赢了强队，比如说此时的巴萨，他相对于巴萨这个概率最高的节点就有了通路，他的排名也就变到了第二。

由于这只球队只在西甲打了一年，出现意外净胜场赢了巴萨是很正常的事情，然而算法却因此考虑了这样的意外情况，这证明pagerank算法并不适用于水平排名的场合。

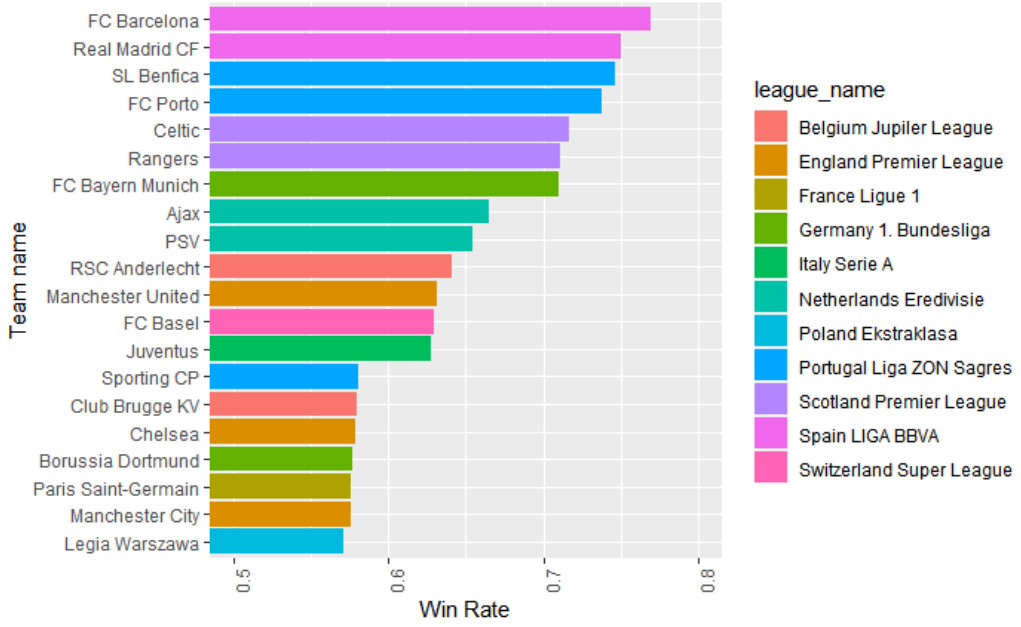
而实际上，我们只使用胜率作为指标进行排行的结果是相当准确的，如下图所示。



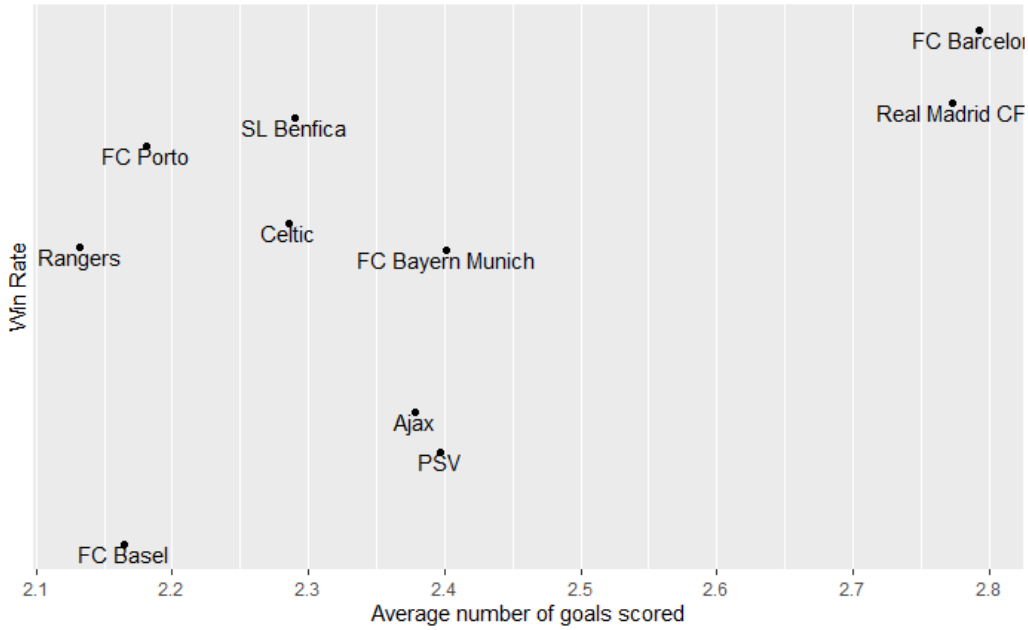
- 全部球队排行

接下来，针对所有联赛的所有球队，经过胜率排序后，前二十名的球队如下，显然排名前二的是西甲的豪门巴塞罗那和皇家马德里，因为西甲其余球队较弱，这两只豪门则获得了非常高的胜率。

然而这样做还是存在一些问题，在排行榜中实际上找到的是各联赛的主宰球队，然而却没有国际米兰、利物浦、马德里竞技等传统强队的身影。弱联赛中排名第一的球队并不一定比的上强联赛中排名靠前的球队，因此我们还应当对球队乘上其对应联赛的水平的权重，才能更好的反应欧洲球队的综合实力。

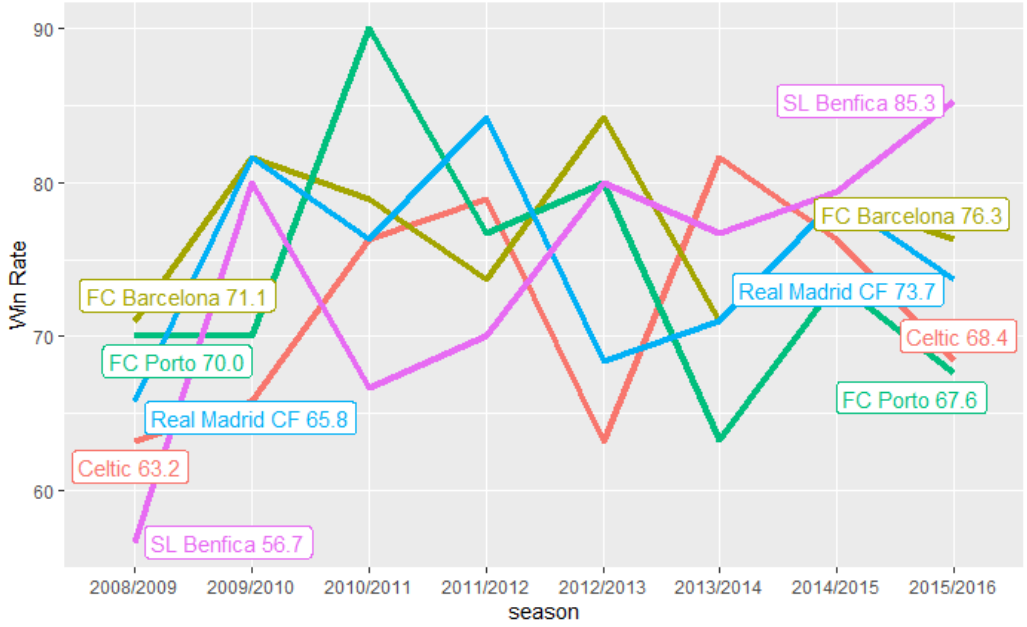


同时对排名前 10 名的球队的场均进球数进行分析，与胜率共同绘制成二维散点图。



发现二者呈正相关关系（虽然相关度并不高，胜率第四的球队 Porto 场均进球数仅排名第 8)，在混淆不清楚的情况下，我们后续决定将净胜球数和历史胜场和败场数一起纳入考虑。

再对前五名的球队各个赛季表现进行分析



发现目前胜率最高的 benfica 在近几年胜率增长迅速，的确本菲卡的势头足够强大直到 2016 年夺得葡甲冠军，而 celtic 有下降趋势。

2.2.2 统计建模

从上述的数据集分析中可以看出，要预测一场比赛结果，**参赛球员的能力**、**两只球队的历史实力**以及**比赛当时的情况**都是需要纳入考虑的。但是由于特征过多，作为初步的尝试，我们决定参考上述描述分析的初步结果，挑选较为简单但全面的特征进行研究。幸运的是，数据集很好的满足了我们的考量，我们的思路如下：

- **球员能力**

在 Player_Attributes 表中有 FIFA 对球员的整体评分，也就是 overall_rating，然而球员的状态是在不断改变的，只使用一个来表示他过去打的所有比赛未免过于僵硬，好在数据集提供了每个球员 2008-2016 年中多个时间点的状态，我们对每一场比赛都选取了距离该场比赛最近的球员状态作为特征，一共 11*2 个。

- **球队历史实力**

  很遗憾，在 Team_Attributes 表中并没有能够整体代表该球队实力的特征。简单对该球队的球员能力进行加权求和是很不合理的，一方面相较于球员能力的特征，这样的线性组合属于冗余信息；另一方面简单的加和没有考虑到球员之间的相互作用与配合。

  在初步思路中暂时不打算处理过于复杂的情况，因此我们决定将球队过去数场比赛的情况作为当前的整体状态进行考虑（毕竟归根结底一场比赛最重要就是要赢球）。同时再考虑一个该球队所属的联赛。

  于是在这里我们使用了描述分析中的思路，生成了多个特征，包括过去 x 场（在代码中采用的是 10 场）分别作为主队和客队的总净胜球数（如果输了就是负的）、过去 x 场分别作为主队和客队的总胜场数和输场数、和这场比赛的对方球队过去 y 场（代码中采用的是 3 场）赢球次数和输球次数、该球队所属的联赛（哑变量）。

- **比赛当时情况**

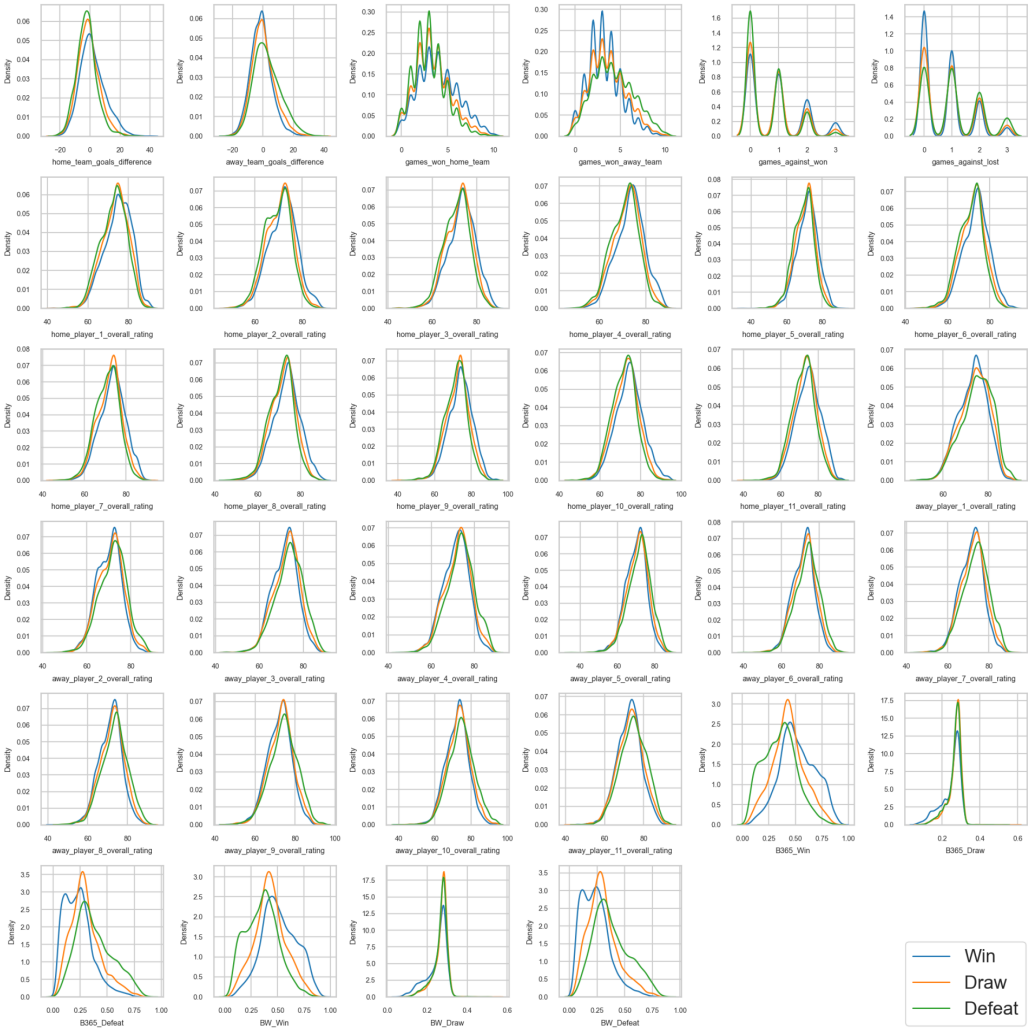
  这个在当前数据是最难考虑的一个地方，因此我们暂时没有将比赛中统计数据进行融合，而是采取了另一个思路。

  我们使用了数据集提供的 10 大博彩网站的赛前赔率数据（为方便起见只用了 Bet365 和 Betway 两个最知名的公司），我们希望赛前的赔率能够包含除开球队、球员数据之外的其他因素，比如赛前的大事件、比赛场地等等因素，从而提高预测胜负关系的效果。

而接下来，则是模型的选用了，作为初步思路我们决定首先选取一些机器学习方法进行研究。同时注意到在特征的选取过程中应当存在不少冗余的情况（赔率数据和前两类数据的关系、球队净胜球数和球队历史胜场情况的关系），我们决定使用一些盲源分离算法首先对数据进行降维处理，再随后再通过机器学习分类器。

那么具体思路如下：

1. 首先将数据集按照上述要求进行特征提取，并随后去除缺失值并进行归一化处理，最终得到 19673 份比赛样本，45 个特征。标签则有 3 类，赢球、平局、输球，比例为 0.46:0.29:0.25。所有特征按照标签进行分类后的密度图见下图。



2. 随后对数据集以 4:1 的比例划分训练集与测试集，而训练集再使用五折交叉验证进行划分。
3. 模型方面的选取分为两个部分，盲源分离算法包括 PCA 和 ICA，分类器包括 RandomForest、AdaBoost、NaiveBayes、KNeighbors、LogisticRegression。将完整的模型定义为"先将原始数据通过一个盲源分离算法，再将输出结果传入分类器进行训练"的管道。并且在最后，我们加入了一个将前述方法全部纳入考虑的 stacking 方法，使用 LogisticRegression 作为最终的分类器进行训练
4. 选择各模型的合理参数范围，将上述所有模型进行网格搜索，使用准确率(Balanced Accuracy)和 F1 分数(f1_score)作为评价指标，以得到最佳的降维方法和分类器组合。

2.3 初步结果与调整

初次尝试中，我们的配置如下表所示：

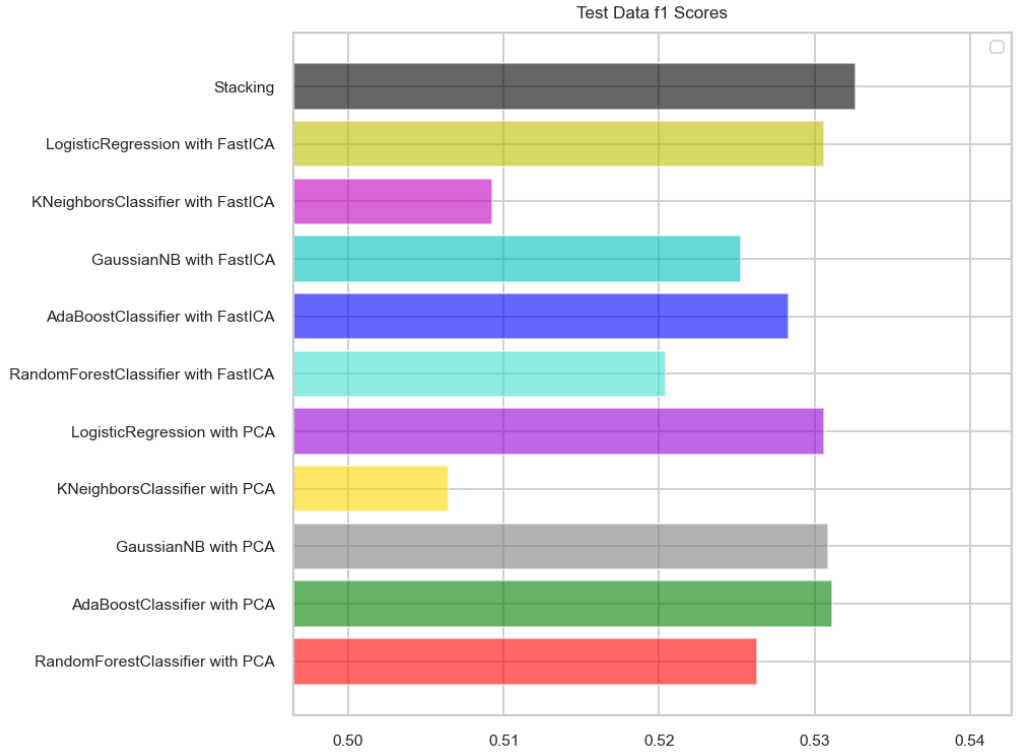
数据集配置		网格搜索配置		
数据种类	**比例**	**模型种类**	**配置**	
训练数据	0.8	PCA&ICA	**n_component**	arange(5,46,8)
测试数据	0.2	RandomForest	**max_features**	[auto、log2]
交叉验证配置			**n_estimators**	[50, 100, 200]
打乱次数	**训练验证比例**	AdaBoost	**learning_rate**	linspace(0.5,2,5)
5	4:1		**n_estimators**	[50, 100, 200]
		NaiveBayes	-	
评价指标		KNeighbors	**n_neighbors**	[3, 5, 10]
balanced accuracy	f1 score	LogisticRegression	**C**	logspace(1,1000,5)

考虑到数据不均匀的问题

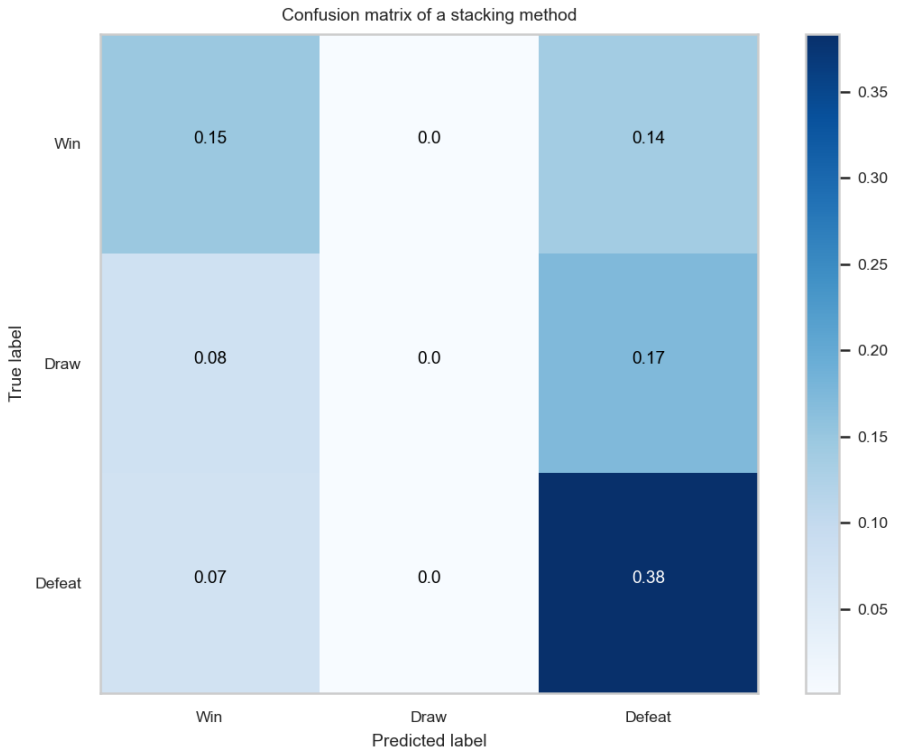
1. 我们采用了分层交叉验证(StratifiedShuffleSplit)，使得交叉验证中的训练集和测试集标签比例适中。

我们使用了 scikit-learn 中自带的 balanced_accuracy_score 针对不平衡的样本使得准确率指标更有意义，而我们的网格搜索使用的评价指标则是 f1_score。

得到的结果如下图所示，结果显示，该情况下使用 PCA 的朴素贝叶斯算法在该情况下效果最好，但也应当注意到，实际上好几个模型之间的差异都不大，这样小的差距可能会受到很多随机影响例如随机数种子的设置，我们认为这和抽取的特征过于简单不够丰富有关，这也是我们希望在下一阶段克服的问题。



最好模型的混淆矩阵如下。



完整的结果如下所示。

	precision	recall	f1-score	support
Defeat	0.49	0.51	0.50	1135
Draw	0.26	0.01	0.01	993
Win	0.55	0.84	0.66	1807
accuracy			0.53	3935
macro avg	0.44	0.45	0.39	3935
weighted avg	0.46	0.53	0.45	3935

3 进一步设想

3.1 动机

针对上述得到的信息，我们提出了可以在接下来进行尝试的内容。

描述分析方面

- 将各场比赛中对应位置的球员维度数据取平均值，根据与平均值的接近程度挑选对应球员，这样应当可以挑选出各个位置具有代表性的球员六维模板。

统计建模方面

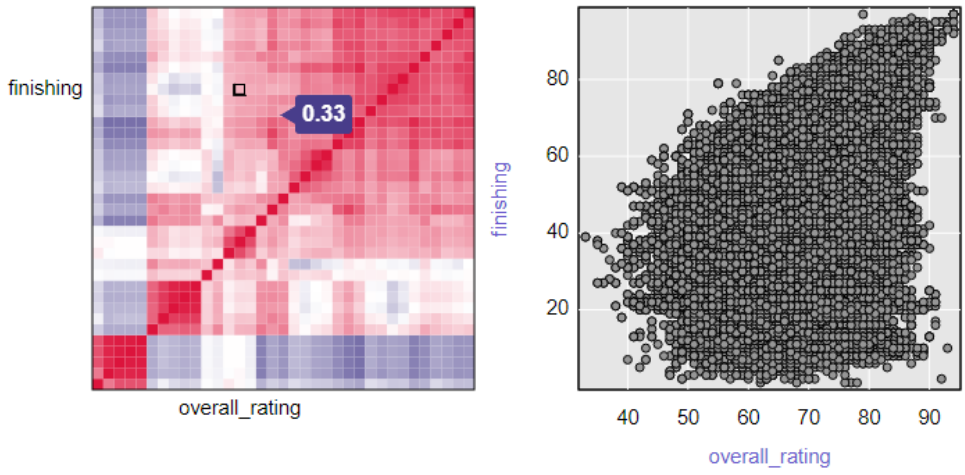
- 在上述描述分析中，我们采用雷达图对球员个人能力进行了描述，这样的六个维度相比于总体评分应当是一个更加全面以及良好的描述，因为可以针对不同位置描述不同情况。因此我们可以将这样 22*6 个特征纳入模型考虑。

- 一个球员可能可以踢多个位置，在一场比赛中他一定踢到了自己最适合的那个位置吗？如果将球队中球员的位置进行调整，代入上面得出的模型，是否可以得到更大的胜率（这里的对手球队可以采用其余球队球员的数据取平均）
- 球队整体数据考虑过于简单（*待补充*）。
- 这样如果采用更多的特征，应当使用深度学习方法和前述的机器学习方法进行对比。
- 在预测比赛胜负时，明星球员的个人能力堆叠和球队整体的配合水平，哪一个对比赛的走向影响更大呢？

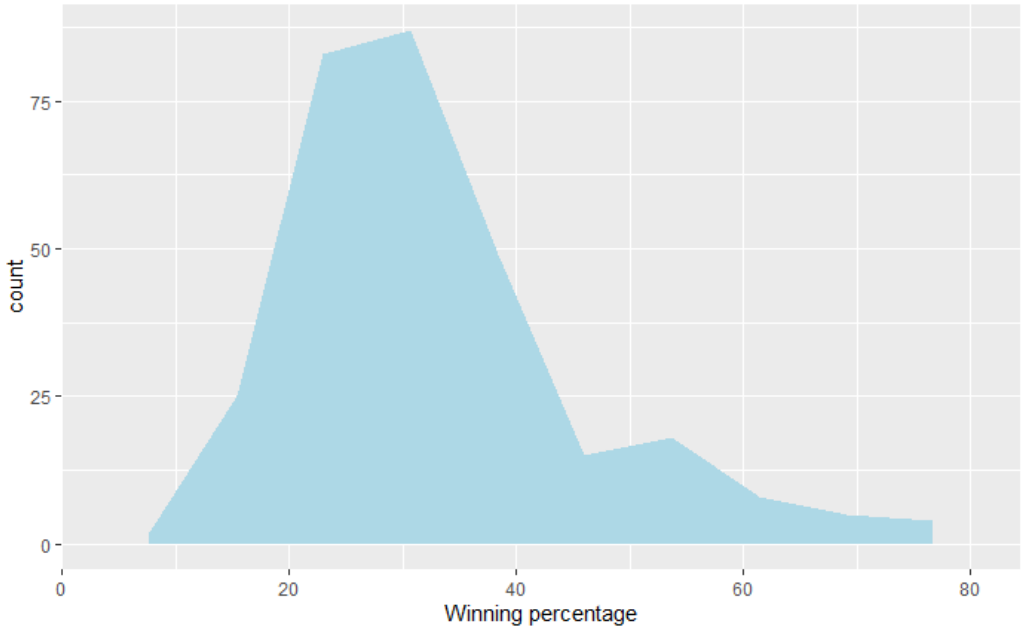
- 我们对球队之间的关系的建模是通过过去数场的净胜球和胜场进行的，过于简单。是否可以采用图神经网络表征球队之间复杂的实力关系呢（比如 A 球队对 B 球队胜率高，对 C 球队胜率低，然而 C 球队对 B 球队胜率高）？
  即使用无向图来对球队之间的胜负关系进行建模，边的权重用两只球队的胜场数除以输场数来表示，边的表征为比赛的情况，节点的表征为球队的实力，这样将预测比赛胜场的问题转化为了链路预测问题。
  但这个思路只能应用于某一个赛季或者将多个赛季分为多个图来表示，和前述内容没有了对比性。

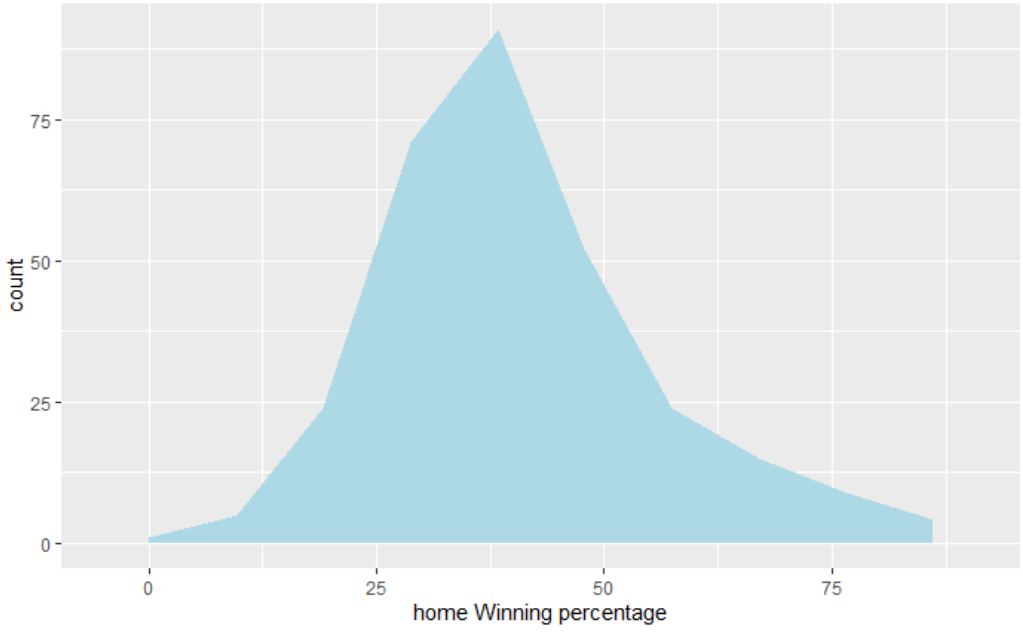
附录

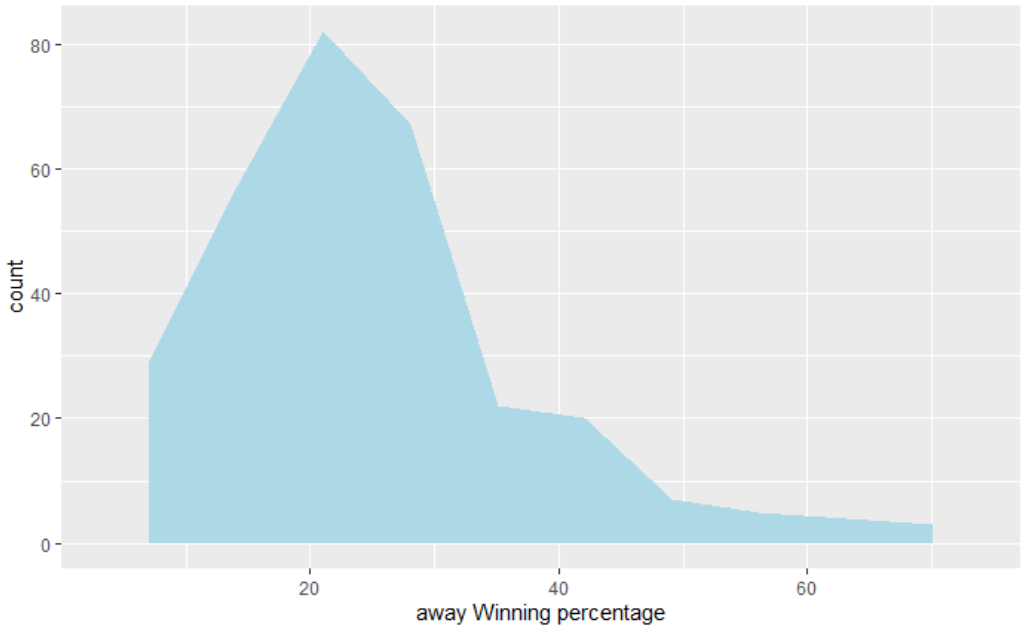
使用相关矩阵对对球员能力各方面数值进行相关性分析，分析球员各方面能力间的关系。



球队胜率情况


count
75
50
25
0
0
20
40
60
80
Winning percentage


count
75
50
25
0
0
25
50
75
home Winning percentage


count
80
60
40
20
0
20
40
60
away Winning percentage