# NBA 球员数据对球队赛季成绩的预测模型


摘要

本文通过综合考虑球员基本数据对球员的整体影响，从而构建衡量球员效率的指标，然后根据球员在整个赛季中的上场时间作为参数，构建所在球队整体球员贡献指标，综合球队的进攻效率，防守效率和净得分等指标，建立对球队在下一赛季胜率的预测模型。本文首先对 NBA 篮球赛制和特点进行总结，分析出影响球队胜率的最主要因素，提取变量特征，建立决策树回归模型，岭回归模型和支持向量回归模型，以下一赛季的数据作为测试集计算出每个模型的均方误差，得到最小的均方误差是多元线性回归模型，利用该模型预测未来赛季中每个球队的胜率。

Abstract

In this article, we comprehensively analysized the basic box scoring, and construct a metric reflecting the contribution of the player to his basketball team, namely the efficiency rate of the player. We use the playing time of the player in the entire season as index to construct the metrics to evaluate the overall contribution of player to each team, combine with other metrics like Net Rating, Offensive Rating, Defensive Ratings etc, to build a model for predicting the winning ratio or a specific team in the next season. We applied ridge regression model, decision tree regression model and supporting vector regression model to predict the next season's winner. We calculate the mean square error as the standard to evaluate which model is the best fit. We use the final model as our result.


## 1. 研究目的

在高级篮球分析领域，研究人员记录和跟踪运动员的身体数据可以制定出更合适的训练方法，对十分有价值的球员可以针对他设计出一个更能够辅助他得到更多分数的球队阵容从而为球队赢得更多的比赛；在娱乐领域，观众可以根据 NBA 官网公布的球员球队数据和信息，计算和预测球队的胜负。本文希望根据个体球员在比赛中的的基本数据得到反映其对所在球队贡献程度的指标，即衡量球员价值的指标，综合某个球队所有球员的贡献，得到球队的整体实力，从而根据球队的整体数据对下一个赛季中球队的胜率进行预测。

## 2. 背景介绍

不久之前，金州勇士队通过分析大量的 NBA 赛事数据和观看大量的球队比赛，找到了防守安东尼·戴维斯的最佳球员。在最新的一赛季比赛中，勇士队的表现有了新的飞跃和突破。今年，几乎 NBA 每一支球队都开始追踪其球员的比赛数据，包括运动员在场上的得分位置，或者每个队员的比赛数据等。通过有效的数据分析，高级的数据模型，和分析工具，NBA 比赛已经逐渐转化为专业篮球比赛。这样的转变不仅影响着球员的打球习惯，教练的训练方法，粉丝与球星的互动方式，甚至在赛制上也有了不同的调整。运动员，教练，粉丝，甚至 NBA 分析员都希望好好利用大量的数据，来满足

他们不同的需求。本文将分析个体球员数据和球队的表现，并运用决策树模型、支持向量机、岭回归来预测球员的成绩。我们获得历史上 20 个赛季的比赛数据，包括个体球员数据和球队比赛数据。

篮球比赛在每个常规赛季中进行 82 场比赛。网上公开了大量的比赛数据，也有很多写得很好的文章和书籍分析球队球员表现。[].但是专业分析员会利用每场比赛球员的走位，和一些不对外公开的数据进行更具体详细的分析。由于数据的有限性，本位会利用网上公开的球员和球队数据进行分析，探索个体球员表现和球队表现的关系，本文最终目的是根据每队球员的具体数据预测 NBA 球队在季后赛的表现。

近几年，衡量球员表现的指标数据在近几年间不断更新，更加全面的反映球员的表现。最基本且最容易获得的数据是每个球员和球队每场比赛的得分数据。得分数据不仅记录某个球员和他队友的各项得分，还记录了他的对手球队和球员的防守数据。之前，得分数据仅仅包含最基础的球员数据，包括球员进球得分，篮板球，助攻数，和投篮命中率等。但这些数据来衡量一个球员的表现，能力和价值是远远不够的。首先，现有的统计量更多偏向记录球员的进攻数据，而忽视了球员的防守贡献水平；如果这个球员在整场比赛中投篮次数更多，助攻更多，这个球员可能是更有价值的球员，但尽管他有很强的的防守能力，比如他多次抢断对方投球，或盖帽，他也不一定能有很高的综合评分。其次，现有的统计数据可能对在场上拥有更多控球权的球员更有利；比如控球后卫将球带到前场并且更多的组织进攻，他带球时间更长，所以他的综合数据可能就更优秀。但是一个球员的控球能力却没有很精准的数据衡量，但控球能力是一个很重要的衡量指标。因此联盟从 1970-1971 年的赛季之后，开始记录球员的犯规数。

至今，由于有了更精准的记录仪器，我们可以获得更准确地球员数据，进行更具体的数据分析。例如金州勇士队，就大量利用大数据分析对球队整体训练计划和决策等方面进行改正。本文不会用到每一个球员在场上的实时位置信息和训练信息等十分具体的数据进行预测，因为这些数据不是对大众公开的。但是本文将会运用更常见的统计指标如射门得分，助攻次数，抢断次数，盖帽次数等，通过分析这些指标对球员和球队贡献的重要性来预测该球队的输赢。

## 3. 数据收集与说明

本文采用的是近 10 个常规赛季的公开数据进行建模。由于比赛从赛制，到规则等各个方面都随着时代不断改变，甚至评判输赢的规则都有改变；尽管目前可以收集到自从 1946-47 赛季开始至今的所有赛事数据，之前的数据对现在比赛的分析也没有太多可借鉴作用。所以我们用今年来的所有常规赛事所公开的数据进行分析与建模。比如在上世纪末期，一个球队的输赢很大程度取决于个头最大的球员（如中风和大前锋），因为当时比赛的节奏相对较慢，主导比赛进程的球员通常是个头大的球员；但是如今，比赛节奏越来越快，比赛进程也更趋于数据化，专业化和技术化，这就为身形较小而灵活，但是技术水平很高，如投篮水平很高，控球能力很强的球员如史蒂芬库里，詹姆斯哈登，凯利欧文等身高不出众但命中率很高的球员。但联盟赛制转变后更有利于进攻型球员，尤其是擅长投射的球员，这使得比赛更激动人心，吸引更多观众的目光。

- 每个赛季所有球员的整体数据，变量包括

以 2011-12 赛季比赛中球员数据为例：

| 变量名 | 变量含义（英文） | 变量含义（中文） |
|---|---|---|
| **Rk** | Rank | 排名 |
| **Pos** | Position | 首发位置 |
| **Age** | Player's age | 年龄 |
| **Tm** | Team | 所在球队 |
| **G** | Games | 该赛季比赛总场数 |
| **GS** | Games Started | *** |
| **MP** | Minutes Played | 球员上场时长 |
| **FG** | Field Goals | 球员投篮命中 |
| **FGA** | Field Goals Attempts | 球员投篮 |
| **FG%** | Field Goal Percentage | 场均球员命中率 |
| **3P** | 3-Point Field Goals | 场均 3 分球命中得分次数 |
| **3PA** | 3-Point Field Goal Attempts | 场均 3 分球投篮次数 |
| **3P%** | FG% on 3-Pt FGAs | 场均 3 分球得分率 |
| **2P** | 2-Point Field Goals | 场均 2 分球命中得分次数 |
| **2PA** | 2-point Field Goal Attempts | 场均 2 分球投篮次数 |
| **2P%** | FG% on 2-Pt FGAs. | 场均 2 分球得分率 |
| **eFG%** | Effective Field Goal Percent | 场均有效的投篮得分率 |
| **FT** | Free Throws | 场均罚球得分 |
| **FTA** | Free Throw Attempts | 场均罚球投篮次数 |
| **FT%** | Free Throw Percentage | 场均罚球命中率 |
| **ORB** | Offensive Rebound | 场均进攻篮板球次数 |
| **DRB** | Defensive Rebounds | 场均防守篮板球次数 |
| **TRB** | Total Rebounds | 场均篮板球总次数 |
| **AST** | Assists | 场均助攻次数 |
| **STL** | Steals | 场均盖帽次数 |
| **BLK** | Blocks | 场均抢断次数 |
| **TOV** | Turnovers | 场均失误次数 |
| **PF** | Personal Fouls | 场均个人犯规次数 |
| **PTS** | Points | 场均得分 |

- 收集每个赛季所有球队的信息如下：

以 2011-12 赛季比赛中球员数据为例：

| 变量名 | 英文名称 | 中文解释 |
|---|---|---|
| Rk | Rank | 球队在赛季中的排名 |
| W | Wins | 球队整赛季赢过的比赛 |
| W/L% | Win-Loss Percentage | 球队的胜负率 |
| MOV | Margin of Victory | 输赢球队比分之差 |
| Ortg | Offensive Rating | 每 100 次进攻的得分 |
| DRtg | Defensive Rating | 每 100 次进攻的失分 |
| NRtg | Net Rating | ·每 100 次进攻机会的净胜分 |
| MOV/A | Adjusted Margin of Victory | 根据对手进攻节奏调整后的 MOV |
| Ortg/A | Adjusted Offensive Rating | 根据对手进攻节奏调整后的每 100 次进攻得分 |
| DRtg/A | Adjusted Defensive Rating | 根据对手进攻节奏调整后的失分 |
| NRtg/A | Adjusted Net Rating | 根据对手进攻节奏调整后的净胜分 |

- 收集每个球队的整个赛季中与之对抗的球队的比赛数据，变量名如表一。
- 每个球队在整个赛季中数据统计，变量名如表一。

## 4. 指标设计

最近，一些体育网站如（虎扑体育），NBA 官网等开始利用简单指标之间的组合，计算出更有意义的指标，使用这些调整后的指标，对分析不同的队伍和不同风格的球员有更显著的意义。比如，一个在更偏防守的球队打球的球员的进攻数据，可能不会比一个在更偏进攻的球队打球的球员的进攻数据更好。然而这并不意味，第一个球员的进攻能力不如第二个球员强。为了削弱球队风格对球员数据的影响，分析者通过不同的方式将数据标准化。其中一种方式，就是根据控球时间标准化球员的各项数据。即根据每个球员每次控球时所产生的数据，可以更加精准的评判一个球员真正的实力。

因为我们的模型是找出球队表现和球员个统计量的 关系，我不仅会用到个体球员数据，也会用到球队统计数据。对于球队数据，我将用到球队的胜率为因变量来衡量球队在整一个赛季的表现，对于球员，我将用根据球员的基本数据构造全面衡量球员效率的霍林格 PER，本模型中 PER 是十分重要的一个指标。

- Team_PER:

1. in_PER:一个十分常见并且广泛应用的指标是 PER(PLAYER EFFICIENCY RATING)，球员效率，根据球员的基本数据生成的衡量球员能力的指标。PER 是体育学家约翰·霍林格最初构建的。这个指标衡量的是球员每分钟的表现和根据球队节奏调整后的表现。这样在节奏慢的球队打球的球员不会因为其球队的节奏而表现出比其他同水平球员更差的数据。此外，PER 是衡量每个球员每分钟表现的数据，这样就不会因为每个球员上场时间不同而有不同的数据表现。因为 PER 指标是各大数据网站公开的，在接下来的模型建立中，将直接使用这个数据。

$$\mathbf{PER_{in}} = \left(\frac{\mathbf{1}}{\mathbf{MP_{in}}}\right) * \left( \mathbf{3P_{in}} + \left(\frac{\mathbf{2}}{\mathbf{3}}\right) * \mathbf{AST_{in}} + \left( \mathbf{2 - factor} * \left(\frac{\mathbf{AST_{team}}}{\mathbf{FG_{team}}}\right) * \mathbf{FG_{in}} + \left( \mathbf{FT_{in}} * \mathbf{0.5} * \left( \mathbf{1} + \left( \mathbf{1} - \left(\frac{\mathbf{AST_{tm}}}{\mathbf{FG_{team}}}\right)\right) + \left(\frac{\mathbf{2}}{\mathbf{3}}\right) * \left(\frac{\mathbf{AST_{team}}}{\mathbf{FG_{team}}}\right)\right)\right)\right) - \mathbf{VOP} * \mathbf{TOV_{in}} * \mathbf{DRB\%} * (\mathbf{FGA_{in}} - \mathbf{FG_{in}}) - \mathbf{VOP} * \mathbf{0.44} * (\mathbf{0.44} + (\mathbf{0.56} * \mathbf{DRB\%})) * (\mathbf{FTA_{in}} - \mathbf{FT_{in}}) + \mathbf{VOP} * (\mathbf{1} - \mathbf{DRB\%}) * (\mathbf{TRB} - \mathbf{ORB}) + \mathbf{VOP} * \mathbf{DRB\%} * \mathbf{ORB} + \mathbf{VOP} * \mathbf{STL_{in}} + \mathbf{VOP} * \mathbf{DRB\%} * \mathbf{BLK_{in}} - \mathbf{PF_{in}} * \left( \left(\frac{\mathbf{FT_{opp}}}{\mathbf{PF_{opp}}}\right) - \mathbf{0.44} * \left(\frac{\mathbf{FTA_{opp}}}{\mathbf{PF_{opp}}}\right) * \mathbf{VOP}\right)\right)$$

其中：

$$\text{factor} = \left(\frac{2}{3}\right) - (0.5 * (AST_{opp})/(FG_{opp}))/(2 * (FG_{opp}/FT_{opp}))$$

$$\text{VOP} = \text{PTS}_{\text{opp}}/(FGA_{opp} - ORB_{opp} + TOV_{opp} + 0.44 * FTA_{opp})$$

$$\text{DRB\%} = (\text{TRB}_{\text{opp}} - \text{ORB}_{\text{opp}})/\text{TRB}_{\text{op}}$$

PER 指标的优缺点：

i. 优点：PER 指标充分运用了球员基本数据的特征，该指标比网上所有公开的原始数据更全面和精准的衡量每场比赛球员的得分情况，从而反映该球员的效率。

ii. 缺点：一个最显著的缺点是，该指标在球员防守指标上缺乏体现。上述公式主要反应球员进攻效率，尽管公式中包含球员的盖帽和抢断数据，但这些数据还是相对片面。

2. team_possesions:每队的的进攻次数，从一个球队拿到控球权算做一次进攻开始，当球队因失误将球丢失或者投篮命中或者被抢断和盖帽，算做一次进攻的停止。在一场比赛中，两队的进攻次数大致相同，所以利用进攻次数可以更好的衡量球队的优秀程度。为了赢得比赛，每支球队应该在进攻中取得最多的得分。

$$\text{Possesion}_{\text{team}} = (FG_{team} + \gamma FT_{team}) + \alpha[(FGA_{team} - FG_{team}) + \gamma(FTA_{team} - FT_{team}) - ORB_{team}] + (1-\alpha)DRB_{opp} + TOV_{team}$$

3. team_pace:每 48 分钟的球队的进攻效率。

$$\text{Pace}_{\text{team}} = \frac{possesions_{team}}{48}$$

4. pace_adjustment：根据比赛双方进攻强度进行调整；

$$\text{Pace}_{\text{ad}} = Pace_{opp}/Pace_{team}$$

5. aPER: 调整后的 PER：$\text{aPER} = \text{pace}_{\text{ad}} * uPER$

6. team_PER:衡量球队打球效率的指标，根据球员 aPER 构成球队 team_PER.

1) 按照每支球队整个赛季中各个球员在场时间将球员排序
2) 将每个球员的 PER 和上场时间相乘
3) 将每队上场时间最长的 12 个球员的 PER 与上场时间相乘后的指标相加

该计算方法的意义：

i. 由于 PER 是一个以分钟为单位的指标，通过与球员的上场时长相乘可以估计出

球员在整个赛季中对球队贡献。

ii. 一些球员通过选秀等方式会被交易到其他球队，所以要计算球员上场时间最长的 12 个球员，更充分可以反映球员对球队的贡献。因为这 12 个球员是球队中比较稳定的中流砥柱.

iii. 该计算方法也可以自动给 PER 更高的球员赋予更大的权重，因为 PER 更高的球员上场时间更长，而 PER 较低的球员的上场时间会相对较短，所以时长作为权重，可以很好的平衡球员之间的效率。就算是某个 PER 很高的球员在赛季中受伤，他的时长也会限制他对球队的贡献。如史蒂芬库里在 19-20 赛季中因伤暂停比赛半年，尽管他是一个价值很高的球员，他的上场时长受到影响，他对球队的贡献

- Team_Rank：球队排名，依据球队赢得比赛的次数，与球队整个赛季中各个方面的表现得到的整体排名。排名综合考虑球队的进攻，防守，总得分和整体数据得出。
- Adjust_MoV：Margin of Victory 比分差距是用来衡量比赛中获胜队伍和失败队伍比赛分数差距的统计量，参考 MOV 可以快速看出比赛是否激烈，获胜队伍胜利的是否显著，大的比分差距代表比赛中获胜方在比赛中强势压制对手，而小的比分差距说明两个球队水平接近，比赛相当激烈。在本数据集中，MOV 指标是取一个队在整个赛季比赛中球队 MOV 指标的平均值。如果一个球队的 Margin of Victory 指标是正数，说明该球队在整个赛季中表现更好。然而在比赛中由于每个球队的进攻风格不同，需要按照进攻强度来调整对应的 MOV 值。尤其是不同的队伍会根据遇到的对手球队调整进攻步伐和阵容等，所以我们利用更好反映球队强弱的 Adjust_Mov 数据来建模。历史上最大的比分差是 68 分，是 1992 年克利夫兰骑士队和迈阿密热火队的比赛，当时骑士队是最强的球队之一。
- Adjust_Ortg：Offensive Rating 是指球队的进攻效率，由于每场比赛下来，球队的进攻节奏会受到很多因素的影响，包括教练风格，球队首发阵容，球员打球习惯，团队合作，对手球队的节奏等。所以很难用一场比赛的总得分作为其进攻效率的判断。为了方便起见，规定将每一百次进攻得分作为球队的进攻效率，并且根据球队的进攻节奏进行调整。一支进攻效率很高的球队，说明球队的进攻水平越高。根据得分制度，一支球队的进攻效率在每 100 次进攻得 100 分左右。
- Adjust_DRtg: Defensive Rating 是衡量一个球队防守效率的指标。由于在比赛进程中，不仅有进攻，还要考虑防守。根据对方球队的进攻效率不同，每个球队的防守程度也不可同日而语，在一支球队面临 100 次进攻时，对方球队的得分就是本球队的失分。我们采用根据进攻节奏步伐调整后的防守效率，更可以客观的反映球队的防守效率。每支球队在比赛时的平均失分在 100 分左右。失分越高，说明球队的防守水平越低。
- Adjust_NRtg：如果只单纯根据 Offensive Rating 和 Defensive Rating 来衡量一个球队的优秀程度多少有一些偏差，这时需要引入球队净得分来更全面的衡量。Net Rating 是球队每 100 次进攻中得分与失分的差。用来衡量球队风格，和更全面的评判球队的表现。一个球队如果在整个赛季中净得分大于 0，说明进攻性更强。当然球队的净得分和球员的表现有很大的关系。球队的净得分均值是 0.
- Win_ratio:在整个赛季的比赛中，赢得的比赛与参加的比赛之比作为胜率。胜率对衡量球队在下一个赛季的表现有着很重要的作用。可以帮助教练调整训练节奏和步伐，帮助球员在选秀中找到一个更合适自己的队伍，可以帮助粉丝们在赌球中

有一个更好的判断。胜率的平均值在 0.5.

## 5. 描述性统计

建模变量统计

| 变量名 | 均值 | 中位数 | 标准差 | 最大值 | 最小值 |
|---|---|---|---|---|---|
| Win Ratio | 0.50 | 0.54 | 0.16 | 0.76 | 0.106 |
| Adjust MOV | 0.00 | 0.75 | 4.75 | 7.43 | -13.96 |
| Adjust Ortg | 105.38 | 105.67 | 3.15 | 111.35 | 96.12 |
| Adjust Drtg | 105.38 | 105.71 | 3.19 | 111.59 | 99.69 |
| Adjust NRtg | 0 | 0.815 | 5.23 | 8.49 | -15.46 |
| Team PER | 76.83 | 69.59 | 38.82 | 152.27 | 14.48 |

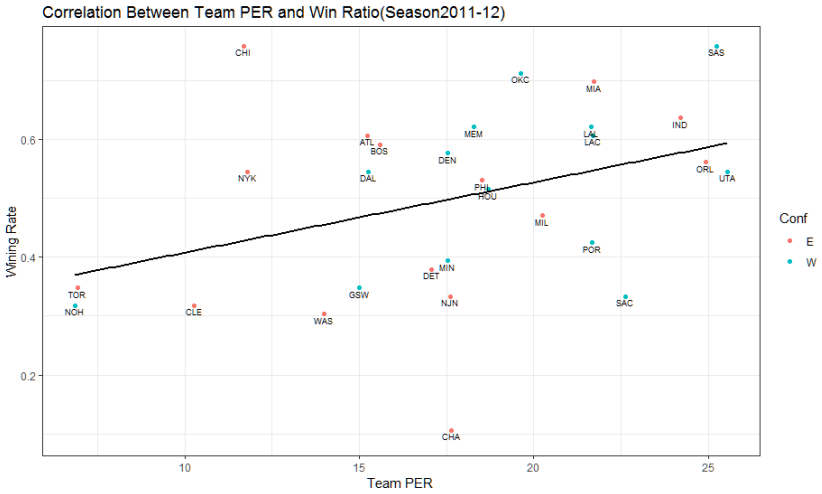



*图 1 Team PER 与 Win Ratio 的关系*

根据公式构建出 Team_PER 指标，球员对球队的整体贡献均值是 76 分，最大贡献可以达到 152 分，最小值不低于 14 分。Team_PER 的标准差为 38 分，体现在数据的离散程度较大。但球员贡献率对球队胜率有正比关系，球员贡献越高，说明球队胜利的可能性越大。

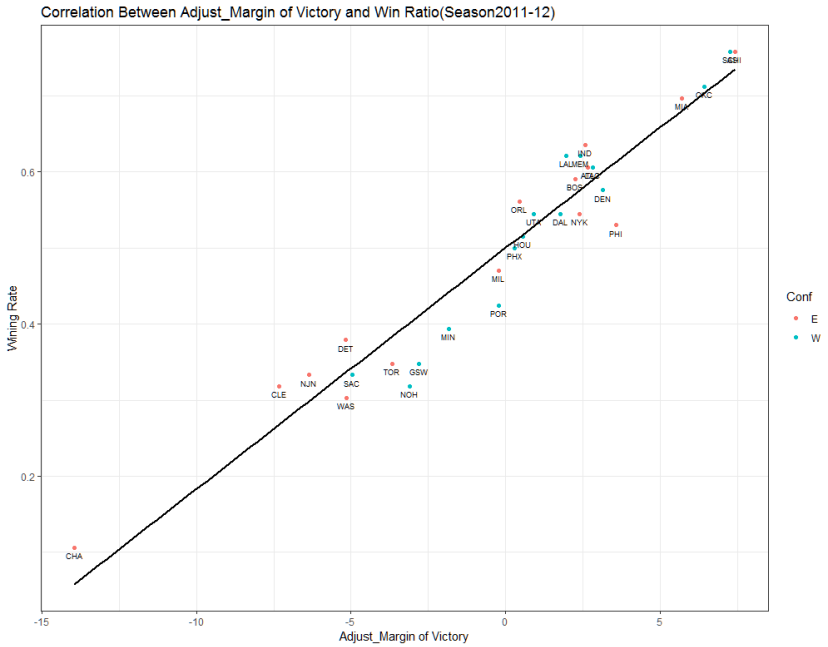



*图 2 MOV 和 Win Ratio 的关系*

Margin of Victory:变量 Margin of Victory 最大值是 7，说明在 11-12 赛季中球队间最大的差距是 7，最小值是-13.96，均值是 0，标准差是 4.75，从上图可知，球队的 MOV 和胜率成正比关系，说明一个球队的 MOV 越高，说明他赢的概率越高。

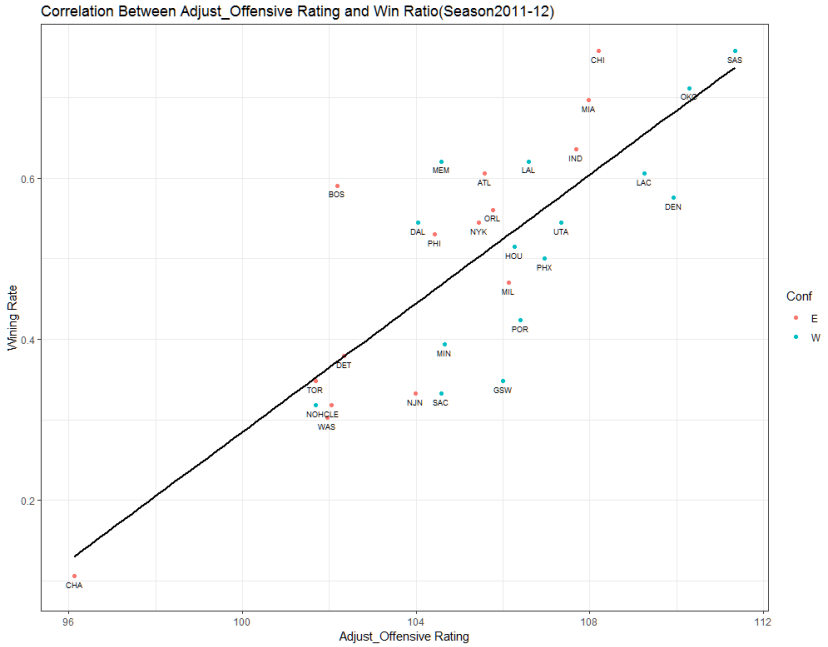



*图 3 Offensive Rating 和 Win Ratio 的关系*

Adjust Offensive Rating:变量 Offensive Rating 最小值是 96.12 分，最大值是 111.35 分，说明在 100 次进攻中一个球队平均可得高达 111 分，最少也不会低于 96 分。均值是 105 分，标准差是 3 分。说明数据之间相差不大。根据上图可以得知，进攻得分越高，胜率越高。

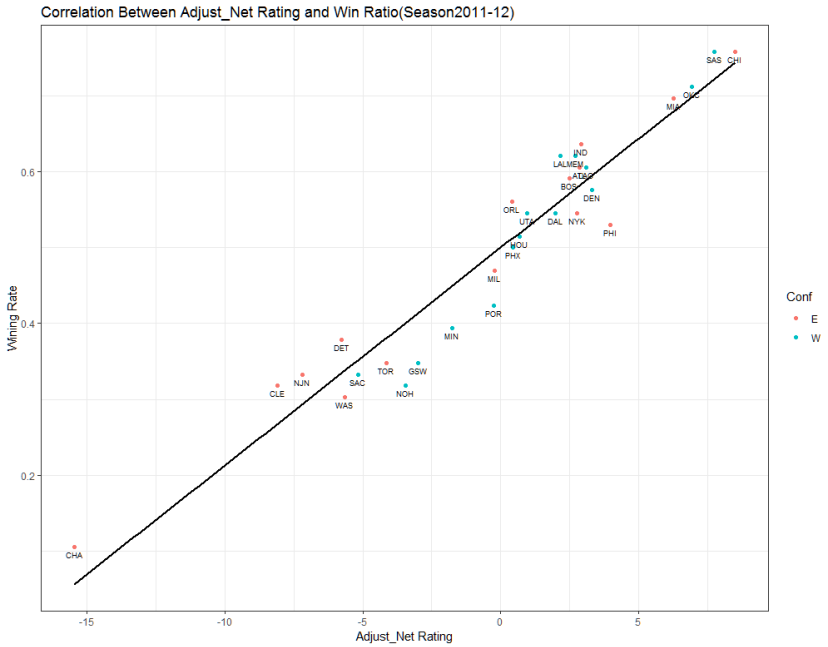



*图 4 Net Rating 和 Wining Ratio 的关系*

Adjust NRtg：变量 Net Rating 最小值是-15 分，最大值是 8 分，说明在一整个赛季中，一个球队每 100 次进攻的平均分差不高于 8 分，不低于 15 分。均值是 0 分，标准差是 5 分。说明数据之间相差不大。根据上图可以得知，净得分越高球队赢得比赛的可能性越高。

## 6. 模型建立

### 6.1 多元回归模型

将标准化以后的变量投入模型，得到如下图结果。

$$f(x) = \text{Intercept} + \beta_1 * Team\ PER + \beta_2 * Rank + \beta_3 * Adjust\ Mov + \beta_4 * Adjust\ ORtg + \beta_5 * AdjustDRtg + \beta_6 * Adjust\ NRtg$$

| | Estimate | Std. Error | t value | Pr(>\|t\|) | |
|---|---|---|---|---|---|
| (Intercept) | 0.002275 | 0.05071 | 0.045 | 0.9646 | |
| Team_PER | 0.113449 | 0.064377 | 1.762 | 0.0919 | . |
| Rk | -0.109022 | 0.176656 | -0.617 | 0.5435 | |
| A_MoV | 2.152315 | 2.187677 | 0.984 | 0.3359 | |
| A_ORtg | -32.308 | 50.818751 | -0.636 | 0.5315 | |
| A_DRtg | 32.674673 | 51.48191 | 0.635 | 0.5322 | |
| A_NRtg | 52.202229 | 84.394901 | 0.619 | 0.5426 | |

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

F-statistic: 62.14 on 6 and 22 DF, p-value: 1.128e-12

根据 F 检验，得到 P 值小于 0.0001，得到整体模型是高度显著的，说明模型中至少一个自变量对因变量有显著影响。且判决系数 R^2 是 0.9，说明模型中自变量可以在很大程度解释因变量。说明我们选取的自变量涵盖大量因变量的信息。但由于每一个自变量做自身 t 检验的显著性都相当低，考虑变量之间是否存在多重共线性。
首先做多重共线性检验，结果如下：

| Team PER | Rank | A_MoV | A_ORtg | A_DRtg | A_NRtg |
|---|---|---|---|---|---|
| 1.58 | 12.3 | 1890 | 101000 | 104000 | 2810000 |

可以看出除了 Team PER 之外，其他变量的 VIF 都非常大，Rank 的方差膨胀因子达到 12，其他变量都超过 200，考虑使用岭回归，消除多重共线性。

岭回归：使用岭回归参数 0.02 时得到最佳的岭回归模型。此时所有变量都是显著的。

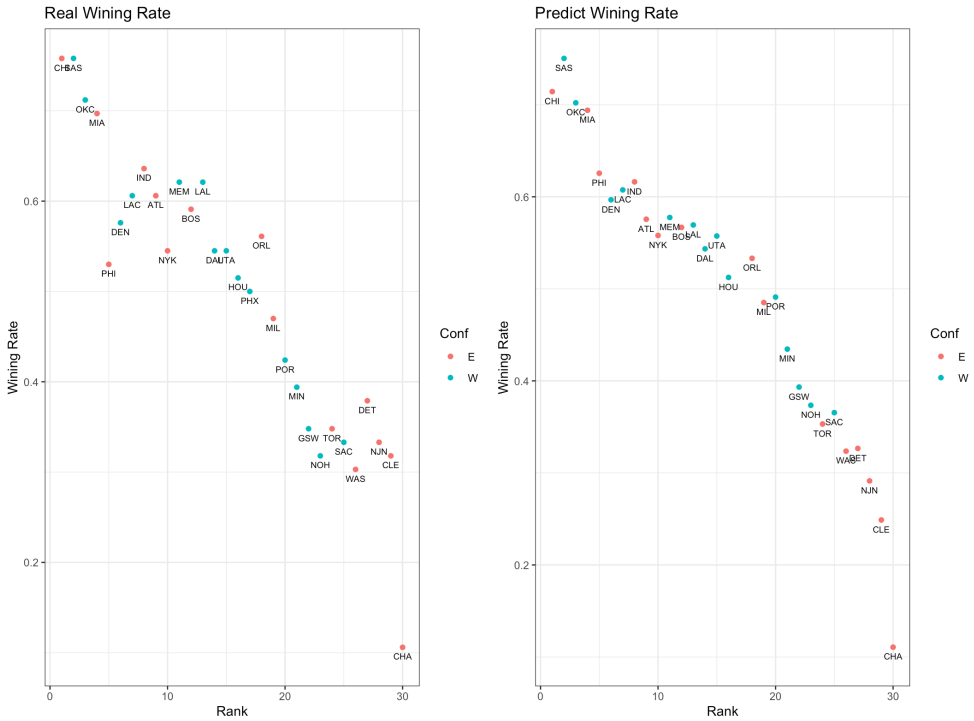
| | Estimate | Std. | t | value | p value | |
|---|---|---|---|---|---|---|
| (Intercept) | 0.0002156 | NA | NA | NA | NA | |
| Team_PER | 0.1235208 | 0.6536105 | 0.3087487 | 2.117 | 0.03426 | * |
| Rk | -0.1900615 | -1.0229824 | 0.6432621 | 1.59 | 0.11177 | |
| A_MoV | 0.2535602 | 1.3653724 | 0.2736496 | 4.989 | 6.05E-07 | *** |
| A_ORtg | 0.1476157 | 0.7913802 | 0.3094264 | 2.558 | 0.01054 | * |
| A_DRtg | -0.1904352 | -1.0232096 | 0.3137362 | 3.261 | 0.00111 | ** |
| A_NRtg | 0.2050112 | 1.1038731 | 0.2137947 | 5.163 | 2.43E-07 | *** |
| Signif. codes: | 0 '***' | 0.001 '**' | 0.01 '*' | 0.05 '.' | 0.1 ' ' | |

Ridge parameter: 0.02080646, chosen automatically, computed using 2 PCs

最终我们得到的的岭回归模型为：

$$\text{Win Ratio} = 0.124 * \text{Team PER} - 0.19 * \text{Rank} + 0.253 * \text{Ad MOV} + 0.148 * \text{Ad ORtg} - 0.19 * \text{Ad DRtg} + 0.21 * \text{Ad NRtg}$$

用此模型对 12-13 赛季的数据进行预测得到如下的结果：




| Team | Real | Team | Predict |
|---|---|---|---|
| SAS | 0.758 | SAS | 0.7503361 |
| CHI | 0.758 | CHI | 0.7142695 |
| OKC | 0.712 | OKC | 0.7020912 |
| MIA | 0.697 | MIA | 0.6939723 |
| IND | 0.636 | **PHI** | 0.6255864 |
| MEM | 0.621 | IND | 0.6162185 |

| | | | |
|---|---|---|---|
| LAL | 0.621 | LAC | 0.6074751 |
| LAC | 0.606 | DEN | 0.596702 |
| ATL | 0.606 | MEM | 0.5774977 |
| BOS | 0.591 | ATL | 0.5756241 |
| DEN | 0.576 | LAL | 0.5693788 |
| ORL | 0.561 | BOS | 0.5667246 |
| UTA | 0.545 | NYK | 0.5579812 |
| NYK | 0.545 | UTA | 0.5573567 |
| DAL | 0.545 | DAL | 0.5434609 |
| PHI | 0.53 | ORL | 0.5331562 |
| HOU | 0.515 | HOU | 0.5123906 |
| PHX | 0.5 | POR | 0.4910005 |
| MIL | 0.47 | MIL | 0.4850674 |
| POR | 0.424 | MIN | 0.4344806 |
| MIN | 0.394 | GSW | 0.3932617 |
| DET | 0.379 | NOH | 0.3734329 |
| TOR | 0.348 | SAC | 0.3654702 |
| GSW | 0.348 | TOR | 0.3531357 |
| SAC | 0.333 | DET | 0.3265933 |
| NJN | 0.333 | WAS | 0.3236267 |
| NOH | 0.318 | NJN | 0.2913074 |
| CLE | 0.318 | CLE | 0.2488394 |
| WAS | 0.303 | CHA | 0.1106624 |

结论：根据上述模型得到的预测结果，可以看到和真实的胜率是差距不大，根据岭回归预测结果，预测值是线性分布的，且与实际结果是一致的。胜率和球队名次是反比关系，名次越高，胜率越低。红色加中的队伍是预测差距超过 10%的队伍。

6.2 支持向量机模型

支持向量机是一种运用凸优化技术对数据集进行二分类的算法。最早起源于 20 世纪 60 年代，在 70 年代得到大量的研究。寻找合适的核函数在 40 年代末就已经被研究过。提高 SVM 的效率，使其应用在大规模数据集上，是统计学家研究的重点。支持向量机是针对二分类问题设计的，多分类任务是基于二分类问题的推广。支持向量回归和支持向量机相似用于回归的算法，所以我们在连续型数据集上使用 SVR 算法。

模型构成：

- 核函数：将低维数据投影到高维数据的函数
- 超平面：在 SVR 中我们将它定义为一条线（蓝色），它将帮助我们预测连续值或目标值
- 间隔：支持向量机会在超平面两边找到两个边界（红色），支持向量可以在间隔上或者外边，这两个边界是用来分开数据的。
- 支持向量：最接近两个间隔的两个（或多个）观测点成为支持向量。最小化支持向量间的距离是算法的核心。在上图中，绿色的点是观测样本，我们的目标是最大化绿色点之间的距离，即最大化间隔。

根据数据的分布和先验知识确定核函数，将无法分割的数据投影到高维空间进行分割。最先尝试径向基函数，即高斯函数进行投射，因为高斯函数适用范围更广，更适合小数据集，且高斯核函数的抗干扰能力强，由于我们数据集存在极端值。

1. 将数据进行标准化，以防止出现因单位不同（有百分数和分数）而产生导致数据之间差距过大而模型不够稳健。
2. 将标准化的 2011-12 年数据作为训练集投入模型 f(x, w, b)，使得预测值 f(x)与 y 之间误差最小，未知参数为 w 和 b。当：

$$|f(x) - y| < \epsilon$$

时不计算损失，即允许 y 与 f(x)之间有$\epsilon$的偏差，当

$$|f(x) - y| \geq \epsilon$$

时计算偏差。即构建一个以 f(x)为中心，2$\epsilon$的间隔带，落入此间隔带的训练样本是被正确预测的样本。

3. 构建超平面方程：$f(x) = w^T \boldsymbol{x} + b \pm \epsilon$

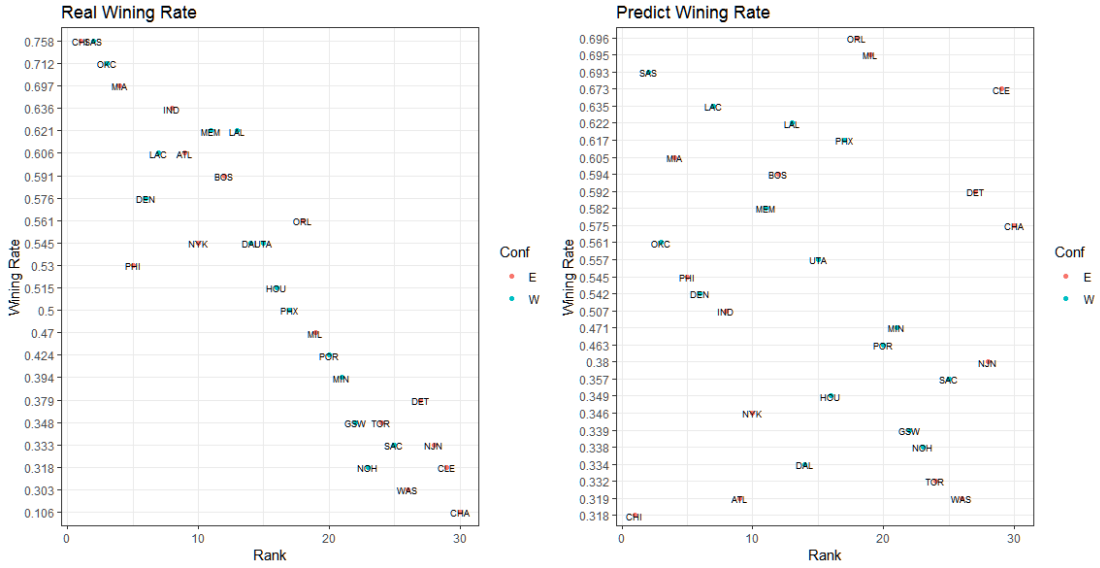
$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m} l_\epsilon(f(x_i) - y_i),$$

其中

$$l_\epsilon(z) = \begin{cases} 0, & if\,|z| < \epsilon \\ |z| - \epsilon & ,otherwise \end{cases}$$

4. 一共输入 30 个队伍的数据，返回 29 个支持向量。

根据该模型对 2012-13 赛季的数据进行预测，得到如下图的预测结果。




根据上述模型得到的预测结果，可以看到和真实的胜率是有一定差距的，根据支持向量模型回归的预测结果点的分散程度更大，没有规律，但是实际上胜率和球队名次是反比关系，名次越高，胜率越低。

| Team | Real | Team | Predict |
|---|---|---|---|
| SAS | 0.758 | ORL | 0.696002 |
| CHI | 0.758 | MIL | 0.6936601 |

| | | | |
|---|---|---|---|
| OKC | 0.712 | SAS | 0.6928794 |
| MIA | 0.697 | CLE | 0.6561883 |
| IND | 0.636 | LAC | 0.6476011 |
| MEM | 0.621 | LAL | 0.6218393 |
| LAL | 0.621 | PHX | 0.6173114 |
| LAC | 0.606 | MIA | 0.6051331 |
| ATL | 0.606 | MEM | 0.6027911 |
| BOS | 0.591 | BOS | 0.5917057 |
| DEN | 0.576 | DET | 0.5831185 |
| ORL | 0.561 | CHA | 0.5751557 |
| UTA | 0.545 | DEN | 0.5609477 |
| NYK | 0.545 | UTA | 0.5607916 |
| DAL | 0.545 | **OKC** | 0.5607916 |
| PHI | 0.53 | PHI | 0.5450222 |
| HOU | 0.515 | IND | 0.5028665 |
| PHX | 0.5 | POR | 0.4655509 |
| MIL | 0.47 | MIN | 0.4558707 |
| POR | 0.424 | NJN | 0.3877971 |
| MIN | 0.394 | HOU | 0.3532919 |
| DET | 0.379 | SAC | 0.3489202 |
| TOR | 0.348 | NYK | 0.3445485 |
| GSW | 0.348 | NOH | 0.3376786 |
| SAC | 0.333 | **DAL** | 0.3339315 |
| NJN | 0.333 | TOR | 0.332214 |
| NOH | 0.318 | GSW | 0.323939 |
| CLE | 0.318 | WAS | 0.3187866 |
| WAS | 0.303 | **ATL** | 0.3187866 |
| CHA | 0.106 | **CHI** | 0.3119168 |

红色加中的队伍是预测差距超过 20%的队伍。预测结果和真实结果相差的百分比不是很大，但是在胜率最高和最低的极端值队伍中，支持向量机模型的表现不够好。会出现将真实值很高的队伍预测出低的结果。

6.3 回归树模型

决策树是一种常见的机器学习方法，其常用于分类问题中，但若将连续变量进行离散化处理后（常用的为二分法），其也可以用于回归问题，此时称决策树为回归树。在将连续变量离散化后，假设所有的输入特征都有有限的离散域，并且有一个称为“分类”的单一目标特征。分类域的每个元素称为一个类。决策树其中每个内部(非叶节点)节点都带有一个输入特性。来自标记了输入特征的节点被标记为目标或输出特征的每个可能值，或者弧导致不同输入特征上的从属决策节点。每片叶子的树是贴上一个类或一个概率分布类,表示数据集已经被树分类到特定的类,或一个特定的概率分布(如果决策树构建良好,是偏向特定子集的类)。

1. 将标准化好的数据投入模型 f(x)中,树模型将按照信息增益最大自动生成结点。信息增益的计算方法如图：

$$\mathrm{E(D)} = -\sum_{t\in T_a} \mathrm{p_k log_2 p_k}$$

$$\mathrm{Gain(D,a)} = \max_{t\in T_a} Gain(D,a,t) = \max_{t\in T_a} \mathrm{E(D)} - \sum_{\lambda\in\{-,+\}} \frac{|D_t^\lambda|}{|D|} E(D^v)$$

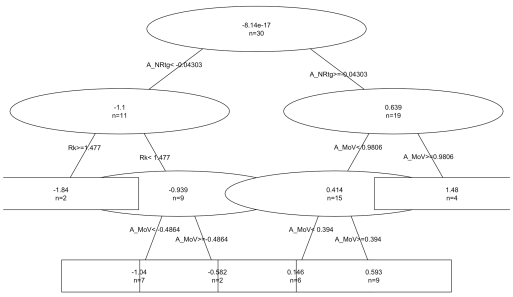
信息增益 gain 越高说明数据中蕴含的信息越多，说明该自变量对因变量的贡献率越高。

2. 如图所示：

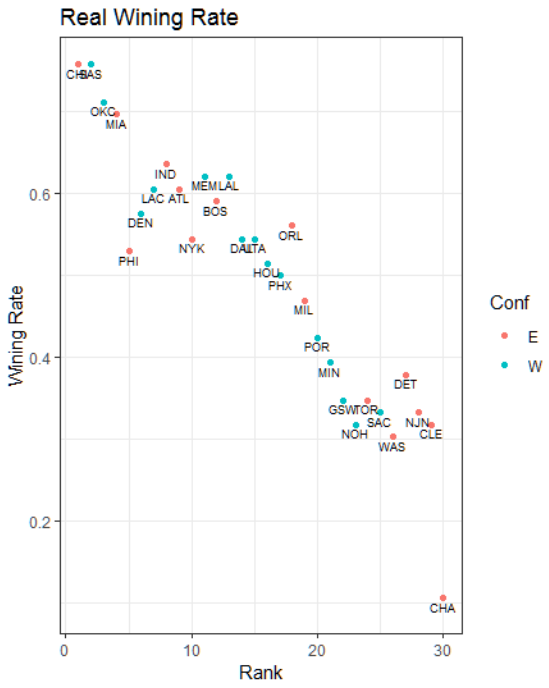
| |
|---|
| 1) root 30 29.00000000 -8.141636e-17 |
| 2) A_NRtg< -0.04303368 11 2.75381800 -1.103319e+00 |
| 4) Rk>=1.476701 2 0.92184250 -1.843950e+00 * |
| 5) Rk< 1.476701 9 0.49111330 -9.387344e-01 |
| 10) A_MoV< -0.4863959 7 0.14577990 -1.040602e+00 * |
| 11) A_MoV>=-0.4863959 2 0.01845982 -5.821989e-01 * |
| 3) A_NRtg>=-0.04303368 19 5.10337900 6.387637e-01 |
| 6) A_MoV< 0.9805847 15 1.38078400 4.139654e-01 |
| 12) A_MoV< 0.3940031 6 0.23847360 1.458166e-01 * |
| 13) A_MoV>=0.3940031 9 0.42327230 5.927313e-01 * |
| 7) A_MoV>=0.9805847 4 0.12202970 1.481757e+00 * |

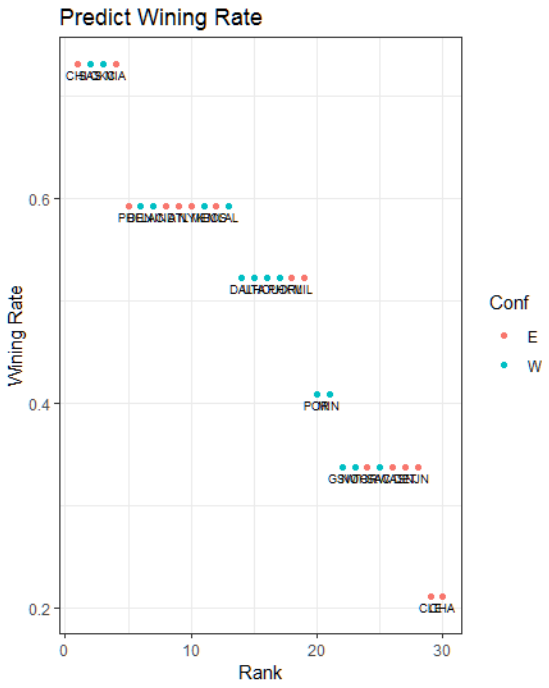
该树的根结点处共有 30 个观测值，第一个节点选择了信息增益最大的 Adjust_NRtg 为第一个划分属性，由于 Adjust_NRtg 是连续型变量，模型计算得到最优分割点为-0.043；因此将数据集划分为大于-0.043 和小于 0.043 的两部分。在第二步分钟共含有 11 个观测，以第二部分小于 0.043 的数据为例，在第二部分数据中继续划分数据，得到第二个节点，选择信息增益最大的 Rank，将数据划分为 Rank>1.48 和小于 1.48 的两部分，第一部分得到了一个叶子节点。将第二部分继续划分下去。最终得到 6 个子节点。树模型如图：

3.




4. 根据该树模型对 12-13 赛季的数据进行预测，返回我们的预测结果如下图：




| Team | Real | Team | Predict |
|---|---|---|---|
| CHA | 0.106 | CLE | 0.2119922 |
| WAS | 0.303 | CHA | 0.2119922 |
| NOH | 0.318 | WAS | 0.3373664 |
| CLE | 0.318 | TOR | 0.3373664 |
| SAC | 0.333 | SAC | 0.3373664 |
| NJN | 0.333 | NOH | 0.3373664 |
| TOR | 0.348 | NJN | 0.3373664 |
| GSW | 0.348 | GSW | 0.3373664 |
| DET | 0.379 | DET | 0.3373664 |
| MIN | 0.394 | POR | 0.4090311 |
| POR | 0.424 | MIN | 0.4090311 |
| MIL | 0.47 | UTA | 0.5226953 |
| PHX | 0.5 | PHX | 0.5226953 |
| HOU | 0.515 | ORL | 0.5226953 |
| PHI | 0.53 | MIL | 0.5226953 |
| UTA | 0.545 | HOU | 0.5226953 |
| NYK | 0.545 | DAL | 0.5226953 |
| DAL | 0.545 | PHI | 0.5924864 |
| ORL | 0.561 | NYK | 0.5924864 |
| DEN | 0.576 | MEM | 0.5924864 |
| BOS | 0.591 | LAL | 0.5924864 |
| LAC | 0.606 | LAC | 0.5924864 |
| ATL | 0.606 | IND | 0.5924864 |
| MEM | 0.621 | DEN | 0.5924864 |
| LAL | 0.621 | BOS | 0.5924864 |
| IND | 0.636 | ATL | 0.5924864 |

| | | | |
|---|---|---|---|
| MIA | 0.697 | SAS | 0.7312879 |
| OKC | 0.712 | OKC | 0.7312879 |
| SAS | 0.758 | MIA | 0.7312879 |
| CHI | 0.758 | CHI | 0.7312879 |

结论：从图中可以看出，回归树模型返回的预测值有 6 个，代表着 6 个叶子节点，每个叶子节点中都包含一些观测值，其预测结果和真实值最为相近。

## 7. 模型比较

根据模型在预测集 12-13 赛季上预测结果和真实值计算出模型的 MSE，结果如下：

$$\mathrm{MSE} = \frac{1}{M}\sum_{m=1}^{M}(y_m - \hat{y}_m)^2$$

| 模型名称 | Mse |
|---|---|
| 支持向量机 | 0.175 |
| 岭回归 | 0.037 |
| 决策树 | 0.039 |

岭回归模型有最小的 MSE 即预测误差最小，但是决策树模型的表现也很好。所以我们可以采用岭回归模型和决策树模型应用于根据球员数据对球队表现进行预测。

## 8. 模型缺点和不可控因素

由于本文试图通过公开的数据和建模寻找球员表现和球队胜负的关系，其中一些不可控因素会导致模型出现一定的问题，例如，一些球员在赛季中受伤退赛或者没有退赛但是伤势会对其表现有很大的影响；或者某些球员在某一赛季中因选秀被交易到其他球队效力；或者某队在赛季中因更换教练导致球队风格有了很大的改变；或者某些球队的数据过于相似而导致其特征不够显著而无法得到更精准的预测。

## 9. 附录

```
setwd("C:\\Users\\Administrator\\Desktop\\NBA\\NBA")
#载入包
library(dplyr)
library(readxl)
library(ggplot2)
library(e1071)
library(gridExtra)
library(DMwR)
library(rpart)
Player_sta <- read_xlsx("2011-12.xlsx",sheet="Player_sta",col_names=T,na="NA")
Team_Name <- read_xlsx("2011-12.xlsx",sheet="Team_Name",col_names=T,na="NA")
Team_Result <- read_xlsx("2011-12.xlsx",sheet="Team_Result",col_names=T,na="NA")
Team_sta <- read_xlsx("2011-12.xlsx",sheet="Team_sta",col_names=T,na="NA")
Opponent_sta <- read_xlsx("2011-12.xlsx",sheet="Opponent_sta",col_names=T,na="NA")
```

```
Team_Result <- right_join(Team_Name,Team_Result,by=c("Full Name"="Full Name"))
Team_sta <- right_join(Team_Name,Team_sta,by=c("Full Name"="Full Name"))
Opponent_sta <- right_join(Team_Name,Opponent_sta,by=c("Full Name"="Full Name"))
Player_sta <- full_join(Team_sta,Player_sta,by=c("Short Name"="Tm"),keep=F)
Player_sta <- full_join(Team_Result,Player_sta,by=c("Short Name"="Short Name"),keep=F)
Player_sta <- full_join(Opponent_sta,Player_sta,by=c("Short Name"="Short Name"),keep=F)
#.x 是 player 的数据，.y 是所在 team 的数据，什么都不加的是对手队伍的数据
#无用的列 Player_sta[,c("Full Name.x","Rk.x.x","Full Name.y","Rk.x")]
#构造数据变量
VOP = Player_sta$PTS.y / (Player_sta$FGA.y - Player_sta$ORB.y + Player_sta$TOV.y + 0.44 *
Player_sta$FTA.y)
DRB_percentage = (Player_sta$TRB.y - Player_sta$ORB.y) / Player_sta$TRB.y
factor = (2/3)-
(0.5*(Player_sta$AST.y/Player_sta$FG.y))/(2*(Player_sta$FG.y/Player_sta$FT.y))
#未调整的 PER
uPER = (1/Player_sta$MP.y)*(Player_sta$`3P.y`+(2/3)*Player_sta$AST.y+
      (2-factor*(Player_sta$AST.x/Player_sta$FG.x))*Player_sta$FG.y+
       (0.5*Player_sta$FT.y*(1+(1-(Player_sta$AST.x/Player_sta$FG.x))+(2/3)*
        (Player_sta$AST.x/Player_sta$FG.x)))-VOP*Player_sta$TOV.y-VOP*DRB_percentage*
      (Player_sta$FGA.y-Player_sta$FG.y)-VOP*0.44*(0.44+(0.56*DRB_percentage))*
       (Player_sta$FTA.y-Player_sta$FT.y)+VOP*(1-DRB_percentage)*(Player_sta$TRB.y-
       Player_sta$ORB.y)+VOP*DRB_percentage*Player_sta$ORB.y+VOP*Player_sta$STL.y+
       VOP*(1-DRB_percentage)*(Player_sta$TRB.y-
Player_sta$ORB.y)+VOP*DRB_percentage*
       Player_sta$ORB.y+VOP*Player_sta$STL.y+VOP*DRB_percentage*Player_sta$BLK.y-
       (Player_sta$FT.y-Player_sta$PF.y)-0.44*VOP*(Player_sta$FTA.y/Player_sta$PF.y))

Play_time<-aggregate(Player_sta$MP.y,by=list(Player_sta$`Short
Name`,Player_sta$Player),sort,decreasing=T)
colnames(Play_time) <- c("Tm","Player","Minutes_Play")
#构建 teamPER
Player_PER <- as.data.frame(cbind(Player_sta$Player,uPER))
colnames(Player_PER) <- c("Player","uPER")
Player_PER <- right_join(Play_time,Player_PER,by=c("Player"="Player"))
Player_PER$uPER <- as.numeric(as.character(Player_PER$uPER))
Player_PER$multi_min <- Player_PER$Minutes_Play*Player_PER$uPER
Team_PER <- aggregate(Player_PER$multi_min,by=list(Player_PER$Tm),sum)
Team_PER <- Team_PER[-which(is.na(Team_PER$x)),]
colnames(Team_PER) <- c("Tm","Team_PER")

Team_Conclusion <- right_join(Team_PER,Team_Result,by=c("Tm"="Short Name"))
#summary(Team_Conclusion)
Team_Conclusion$Conf <- as.factor(Team_Conclusion$Conf)
Team_Conclusion$Div <- as.factor(Team_Conclusion$Div)
```

```
colnames(Team_Conclusion)<-c("Team","Team_PER","Full
Name","Rk","Conference","Div","W",
"L","Wining_Rate","MoV","ORtg","DRtg","NRtg","A_MoV",
 "A_ORtg","A_DRtg","A_NRtg")
#model2 svm
svm1 <-
svm(Wining_Rate~Team_PER+Rk+A_MoV+A_ORtg+A_DRtg+A_NRtg,data=Team_Conclusion
,kernal="sigmoid")
pre_result <-
round(predict(svm1,Team_Conclusion[,c("Team_PER","Rk","A_MoV","A_ORtg","A_DRtg","A
_NRtg")]),3)
Compare <-
as.data.frame(cbind(Team_Conclusion$Team,as.character(Team_Conclusion$Conference),p
re_result,Team_Conclusion$Wining_Rate,Team_Conclusion$Rk))
colnames(Compare)<-c("Team","Conf","Predict","Real","Rank")
Compare$Rank <- as.numeric(as.character(Compare$Rank))
Compare$Predict <- as.numeric(as.character(Compare$Predict))
Compare$Real <- as.numeric(as.character(Compare$Real))
mse <- sqrt(mean((Compare$Predict-Compare$Real)^2))

p1 <- ggplot(Compare,aes(x=Rank,y=Real))+
  geom_point(aes(color=Conf))+geom_text(aes(label=Team),size=2.5,nudge_y=-0.01)+
  labs(title="Real Wining Rate")+
  xlab("Rank")+ylab("Wining Rate")+
  theme(plot.title = element_text(size = 8))+
  theme_bw()
p2 <- ggplot(Compare,aes(x=Rank,y=Predict))+
  geom_point(aes(color=Conf))+geom_text(aes(label=Team),size=2.5,nudge_y=-0.01)+
  labs(title="Predict Wining Rate")+
  xlab("Rank")+ylab("Wining Rate")+
  theme(plot.title = element_text(size = 8))+
  theme_bw()
grid.arrange(p1, p2,ncol=2)

#model3 glm
lm1 <-
lm(Wining_Rate~Team_PER+Rk+A_MoV+A_ORtg+A_DRtg+A_NRtg,data=Team_Conclusion)
summary(lm1)
pre_result <-
round(predict(lm1,Team_Conclusion[,c("Team_PER","Rk","A_MoV","A_ORtg","A_DRtg","A_
NRtg")]),3)
Compare <-
as.data.frame(cbind(Team_Conclusion$Team,as.character(Team_Conclusion$Conference),p
re_result,Team_Conclusion$Wining_Rate,Team_Conclusion$Rk))
colnames(Compare)<-c("Team","Conf","Predict","Real","Rank")
```

```
Compare$Rank <- as.numeric(as.character(Compare$Rank))
Compare$Predict <- as.numeric(as.character(Compare$Predict))
Compare$Real <- as.numeric(as.character(Compare$Real))
mse <- sqrt(mean((Compare$Predict-Compare$Real)^2,na.rm = T))

p1 <- ggplot(Compare,aes(x=Rank,y=Real))+
  geom_point(aes(color=Conf))+geom_text(aes(label=Team),size=2.5,nudge_y=-0.01)+
  labs(title="Real Wining Rate")+
  xlab("Rank")+ylab("Wining Rate")+
  theme(plot.title = element_text(size = 8))+
  theme_bw()
p2 <- ggplot(Compare,aes(x=Rank,y=Predict))+
  geom_point(aes(color=Conf))+geom_text(aes(label=Team),size=2.5,nudge_y=-0.01)+
  labs(title="Predict Wining Rate")+
  xlab("Rank")+ylab("Wining Rate")+
  theme(plot.title = element_text(size = 8))+
  theme_bw()
grid.arrange(p1, p2,ncol=2)
```

## 10. 参考文献


[1]Justin Kubatko, Dean Oliver, A Starting Point for Analyzing Basketball Statistics, Journal of Quantitative Analysis in Sports. Volume 3, Issue 3 2007

[2]Alexander Franks,Andrew Miller, Luke Bornn, Kirk Goldberry, Harward Univeristy
Counterpoints: Advanced Defensive Metrics for NBA Basketball, Mit Sloan, Sports Analyics Conference.

[3]https://www.basketball-reference.com/leagues/NBA_2012.html#all_team-stats-per_poss

[4]https://wenku.baidu.com/view/bb2bb542fe4733687e21aa44.html

[5]Berri,David J., Martin B. Schmidt, and Stacey L.Book. The Wages of Wins: Taking Measure of TheMany Myths in Modern Sports. 2006. Standford University Press, Standard, CA: Stanford University Press

[6]Hollinger John:”What is PER?”
http://sports.espn.go.com/nba/columns/story?columnist=hollinger_john&id=2850240

[7]Fixler, Kevin:”Disapperance of The Traditional NBA Big Man”
http://www.thepostgame.com/blog/eye-performance/201206/nba-centers-big-man-clifford-ray-kareem-ewing-howard-mikan-wilt