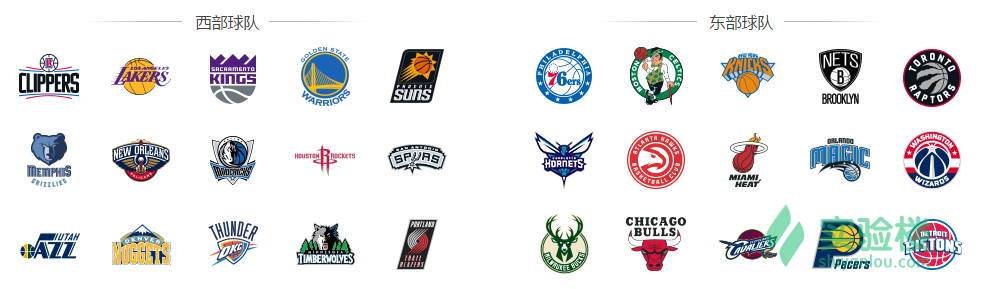
利用Python进行NBA比赛数据分析

## **一、介绍**

### **1.1 内容简介**

不知道你是否朋友圈被刷屏过nba的某场比赛进度或者结果？或者你就是一个nba狂热粉，比赛中的每个进球，抢断或是逆转压哨球都能让你热血沸腾。除去观赏精彩的比赛过程，我们也同样好奇比赛的结果会是如何。因此本节课程，将给同学们展示如何使用nba比赛的以往统计数据，判断每个球队的战斗力，及预测某场比赛中的结果。

我们将基于2015-2016年的NBA常规赛及季后赛的比赛统计数据，预测在当下正在进行的2016-2017常规赛每场赛事的结果。



### **1.2 知识点**

* nba球队的Elo score计算
* 特征向量
* 逻辑回归

### **1.3 环境**

* python2.7
* pycharm

### **1.4 流程**

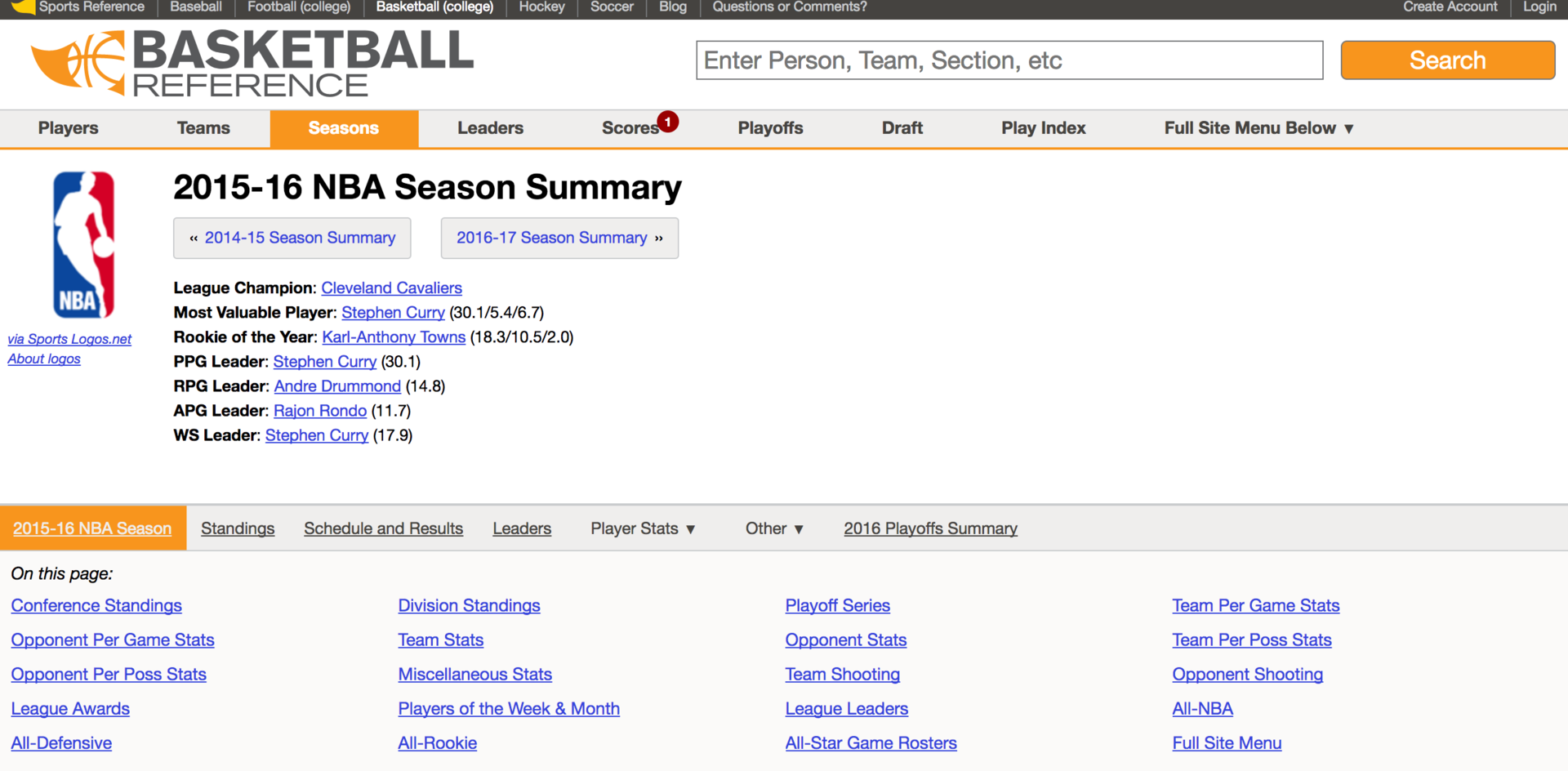
我们将按照下面的流程实现NBA比赛数据分析的任务：

1. 获取比赛统计数据
2. 比赛数据分析，得到代表每场比赛每支队伍状态的特征表达
3. 利用机器学习方法学习每场比赛与胜利队伍的关系，并对2016-2017的比赛进行预测

## **二、获取 NBA比赛统计数据**

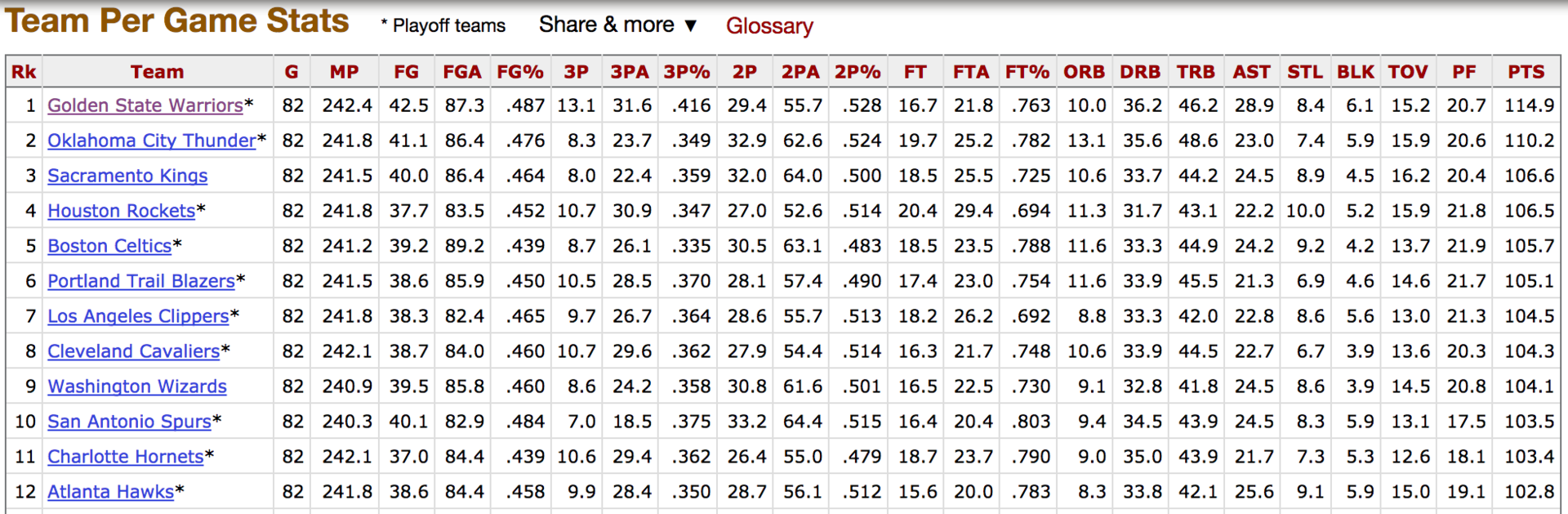
## **2.1 比赛数据介绍**

我们将采用[Basketball Reference.com](http://www.basketball-reference.com/)中的统计数据。在这个网站中，你可以看到不同球员、队伍、赛季和联盟比赛的基本统计数据，如得分，犯规次数等情况，胜负次数等情况。而我们在这里将会使用2015-16 NBA Season Summary中数据。



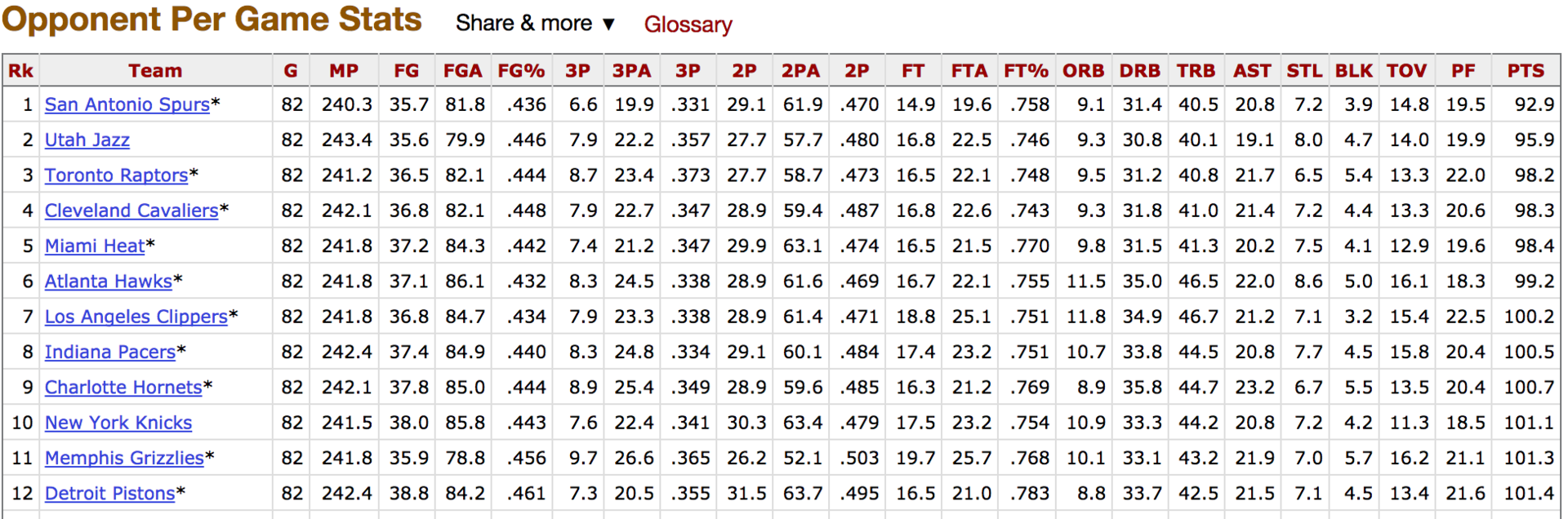
在这个2015-16总结的所有表格中，我们将使用的是以下三个数据表格：

* **Team Per Game Stats**：每支队伍平均每场比赛的表现统计

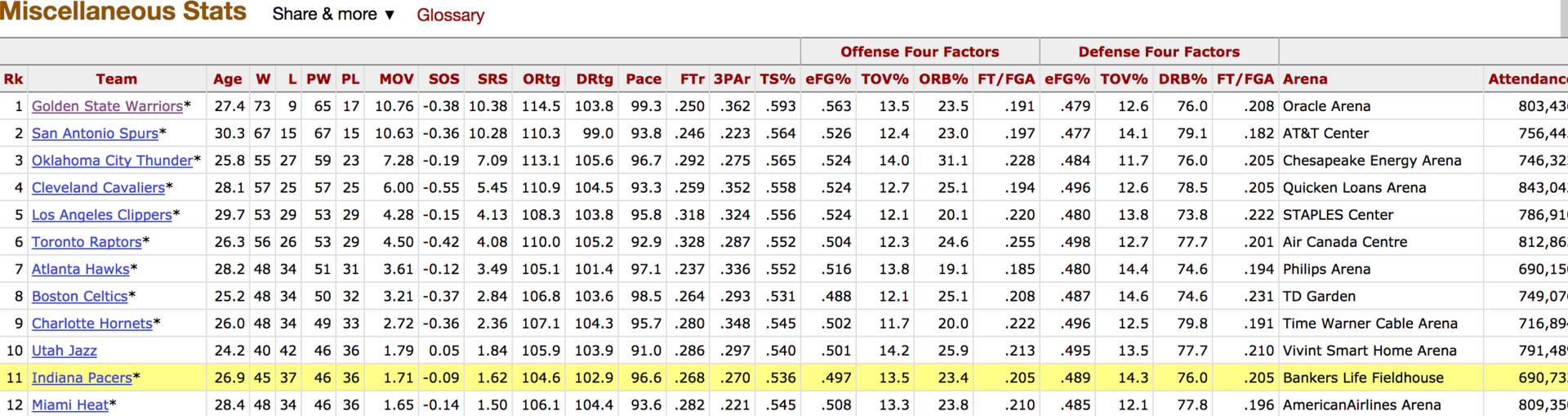


|  |  |
| --- | --- |
| 数据名 | 含义 |
| Rk -- Rank | 排名 |
| G -- Games | 参与的比赛场数（都为82场） |
| MP -- Minutes Played | 平均每场比赛进行的时间 |
| FG--Field Goals | 投球命中次数 |
| FGA--Field Goal Attempts | 投射次数 |
| FG%--Field Goal Percentage | 投球命中次数 |
| 3P--3-Point Field Goals | 三分球命中次数 |
| 3PA--3-Point Field Goal Attempts | 三分球投射次数 |
| 3P%--3-Point Field Goal Percentage | 三分球命中率 |
| 2P--2-Point Field Goals | 二分球命中次数 |
| 2PA--2-point Field Goal Attempts | 二分球投射次数 |
| 2P%--2-Point Field Goal Percentage | 二分球命中率 |
| FT--Free Throws | 罚球命中次数 |
| FTA--Free Throw Attempts | 罚球投射次数 |
| FT%--Free Throw Percentage | 罚球命中率 |
| ORB--Offensive Rebounds | 进攻篮板球 |
| DRB--Defensive Rebounds | 防守篮板球 |
| TRB--Total Rebounds | 篮板球总数 |
| AST--Assists | 助攻 |
| STL--Steals | 抢断 |
| BLK -- Blocks | 封盖 |
| TOV -- Turnovers | 失误 |
| PF -- Personal Fouls | 个犯 |
| PTS -- Points | 得分 |

* Opponent Per Game Stats：所遇到的对手平均每场比赛的统计信息，所包含的统计数据与Team Per Game Stats中的一致，只是代表的该球队对应的对手的



* Miscellaneous Stats：综合统计数据

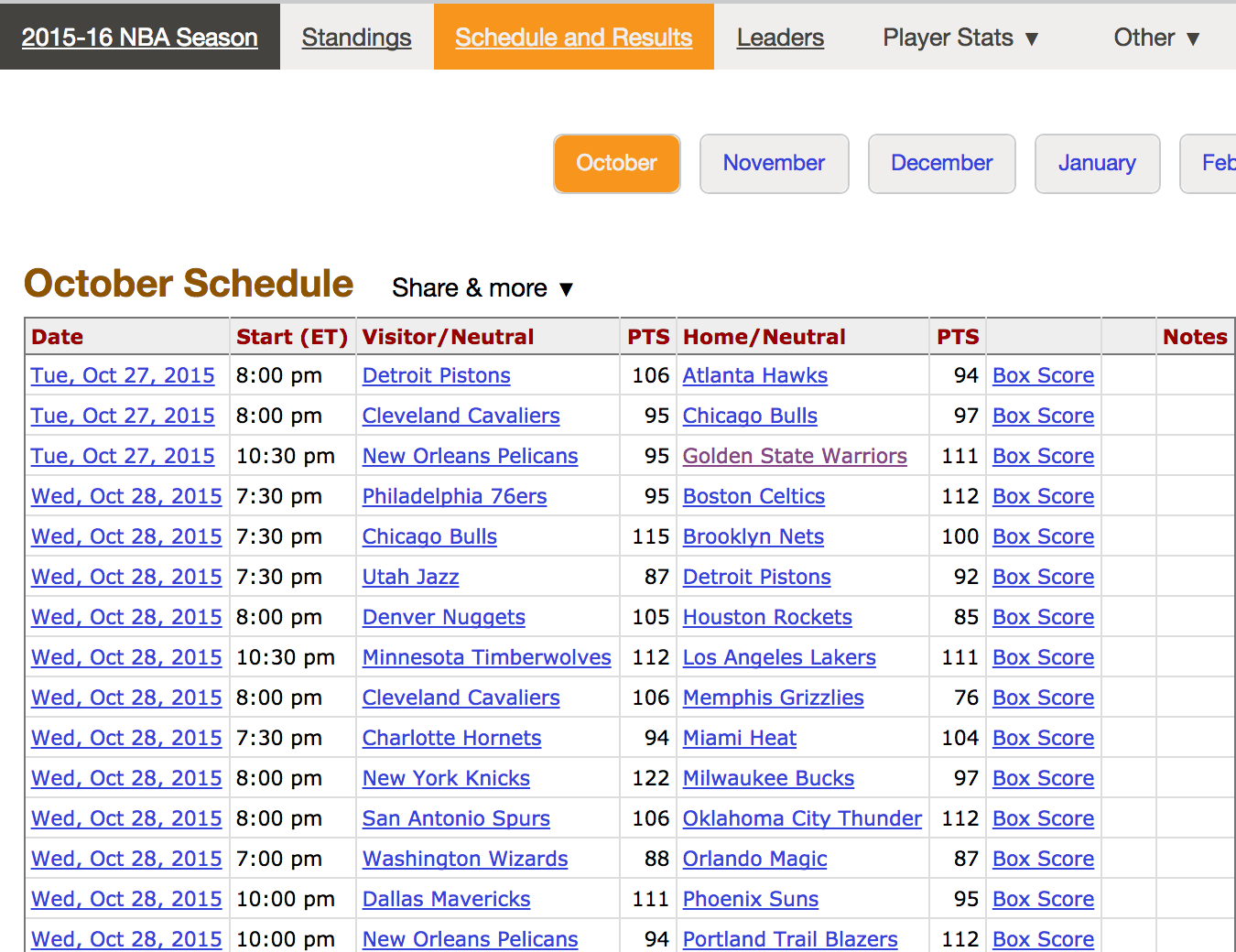


|  |  |
| --- | --- |
| 数据项 | 数据含义 |
| Rk (Rank) | 排名 |
| Age | 队员的平均年龄 |
| W (Wins) | 胜利次数 |
| L (Losses) | 失败次数 |
| PW (Pythagorean wins) | 基于毕达哥拉斯理论计算的赢的概率 |
| PL (Pythagorean losses) | 基于毕达哥拉斯理论计算的输的概率 |
| MOV (Margin of Victory) | 赢球次数的平均间隔 |
| SOS (Strength of Schedule) | 用以评判对手选择与其球队或是其他球队的难易程度对比，0为平均线，可以为正负数 |
| SRS (Simple Rating System) | 3 |
| ORtg (Offensive Rating) | 每100个比赛回合中的进攻比例 |
| DRtg (Defensive Rating) | 每100个比赛回合中的防守比例 |
| Pace (Pace Factor) | 每48分钟内大概会进行多少个回合 |
| FTr (Free Throw Attempt Rate) | 罚球次数所占投射次数的比例 |
| 3PAr (3-Point Attempt Rate) | 三分球投射占投射次数的比例 |
| TS% (True Shooting Percentage) | 二分球、三分球和罚球的总共命中率 |
| eFG% (Effective Field Goal Percentage) | 有效的投射百分比（含二分球、三分球） |
| TOV% (Turnover Percentage) | 每100场比赛中失误的比例 |
| ORB% (Offensive Rebound Percentage) | 球队中平均每个人的进攻篮板的比例 |
| FT/FGA | 罚球所占投射的比例 |
| eFG% (Opponent Effective Field Goal Percentage) | 对手投射命中比例 |
| TOV% (Opponent Turnover Percentage) | 对手的失误比例 |
| DRB% (Defensive Rebound Percentage) | 球队平均每个球员的防守篮板比例 |
| FT/FGA (Opponent Free Throws Per Field Goal Attempt) | 对手的罚球次数占投射次数的比例 |

毕达哥拉斯定律



我们将用这三个表格来评估球队过去的战斗力，另外还需2015-16 NBA Schedule and Results中的2015~2016年的nba常规赛及季后赛的每场比赛的比赛数据，用以评估Elo score（在之后解释）。在[Basketball Reference.com](http://www.basketball-reference.com/)中按照从常规赛至季后赛的时间。列出了2015年10月份至2016年6月份的每场比赛的比赛情况。



可在上图中，看到2015年10月份的部分比赛数据。在每个*Schedule*表格中所包含的数据为：

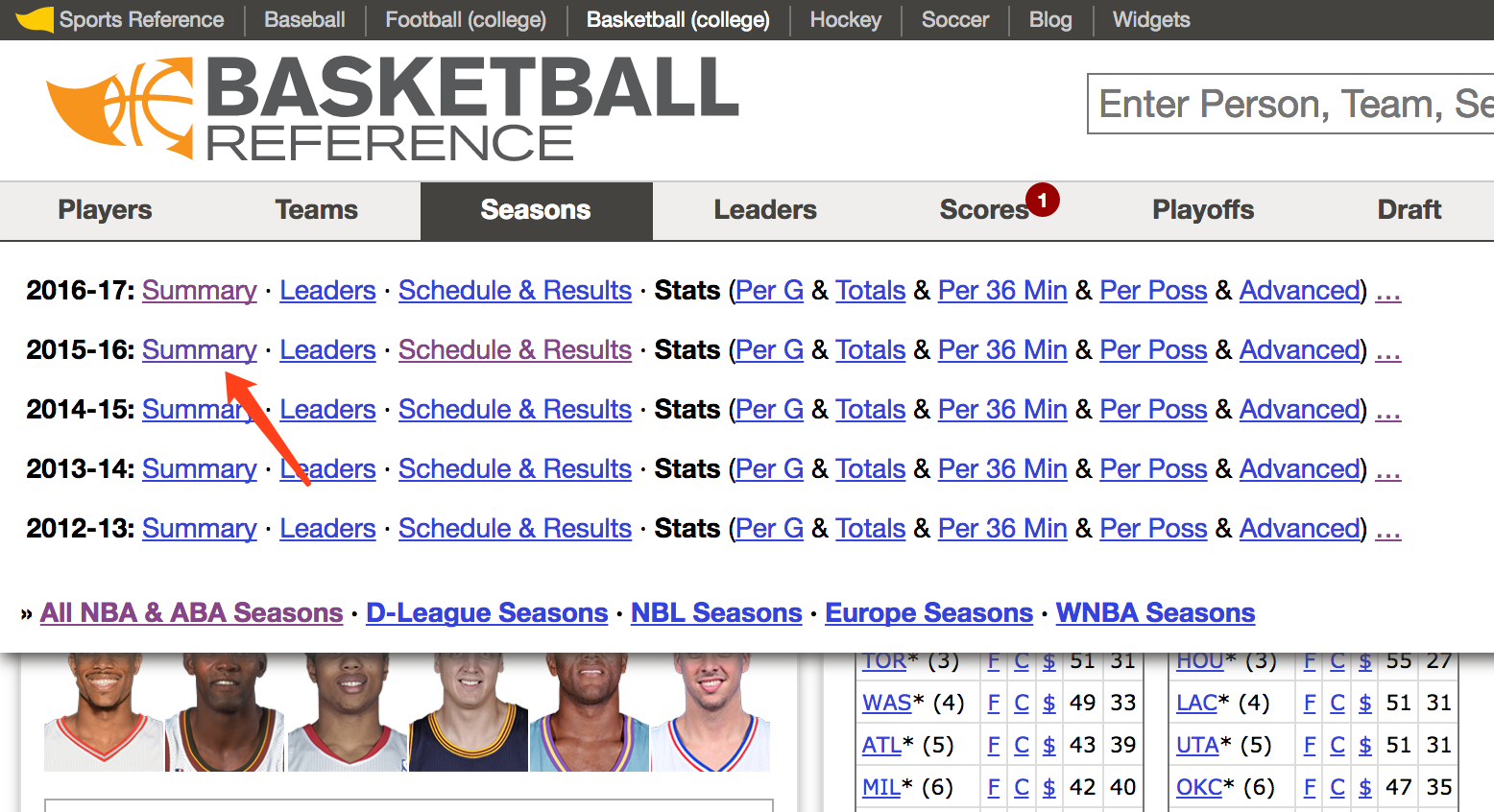
|  |  |
| --- | --- |
| 数据项 | 数据含义 |
| Date | 比赛日期 |
| Start (ET) | 比赛开始时间 |
| Visitor/Neutral | 客场作战队伍 |
| PTS | 客场队伍最后得分 |
| Home/Neutral | 主场队伍 |
| PTS | 主场队伍最后得分 |
| Notes | 备注，表明是否为加时赛等 |

在预测时，我们同样也需要在2016-17 NBA Schedule and Results中2016~2017年的NBA的常规赛比赛安排数据。

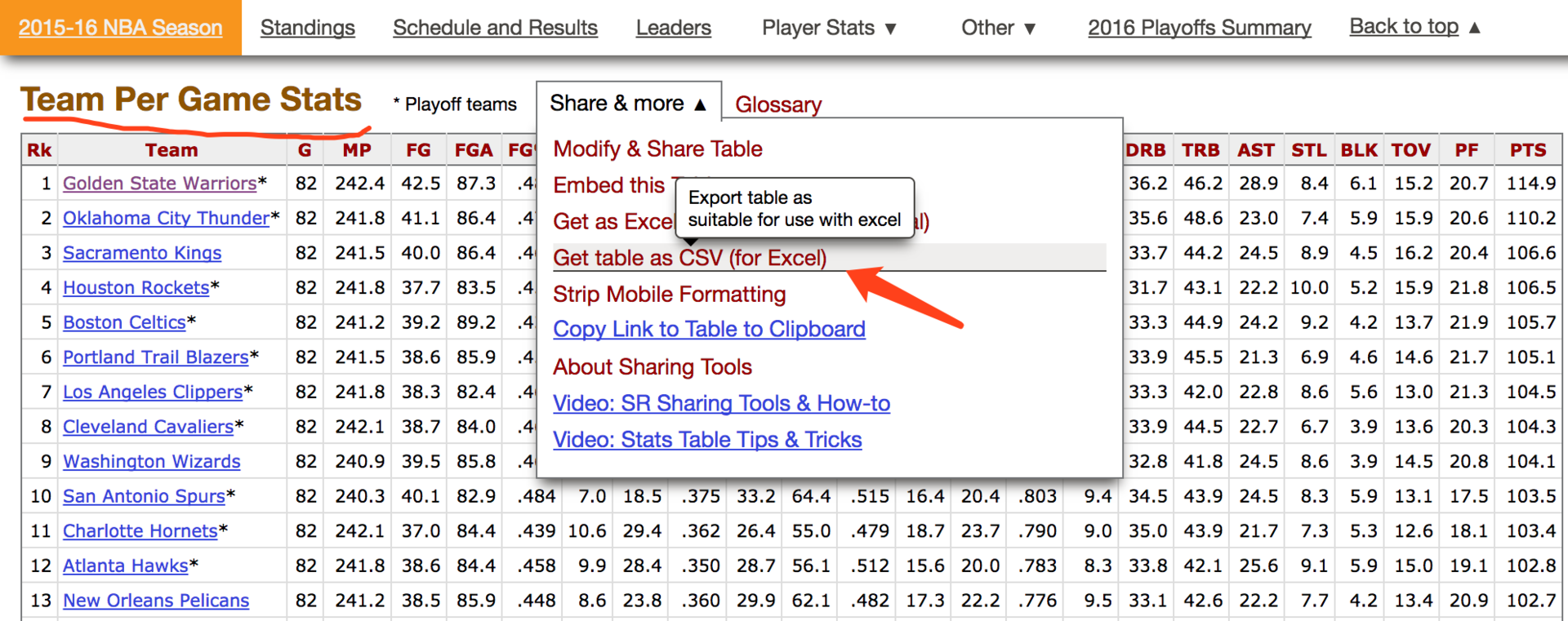
### **2.2 获取比赛数据**

我们将以获取Team Per Game Stats表格数据为例，展示如何获取这三项统计数据。

1. 进入到用[Basketball Reference.com](http://www.basketball-reference.com/)中，在导航栏中选择Season并选择2015~2016赛季中的Summary：



1. 进入到2015~2016年的Summary界面后，滑动窗口找到Team Per Game Stats表格，并选择左上方的Share & more，在其下拉菜单中选择Get table as CSV (for Excel)：



1. 复制在界面中生的的csv格式数据，并复制粘贴至一个文本编辑器保存为csv文件即可。

## **三、数据分析**

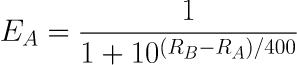
在获取到数据之后，我们将利用每支队伍过去的比赛情况和Elo 等级分来判断每支比赛队伍的可胜概率。在评价到每支队伍过去的比赛情况时，我们将使用到Team Per Game Stats，Opponent Per Game Stats和Miscellaneous Stats（之后简称为T、O和M表）这三个表格的数据，作为代表比赛中某支队伍的比赛特征。我们最终将实现针对每场比赛，预测比赛中哪支队伍最终将会获胜，但并不是给出绝对的胜败情况，而是预判胜利的队伍有多大的获胜概率。因此我们将建立一个代表比赛的特征向量。由两支队伍的以往比赛情况统计情况（T、O和Ｍ表），和两个队伍各自的Elo等级分构成。

关于Elo score等级分，不知道同学们是否看过《社交网络》这部电影，在这部电影中，Mark（主人公原型就是扎克伯格，FaceBook创始人）在电影起初开发的一个美女排名系统就是利用其好友Eduardo在窗户上写下的排名公式，对不同的女生进行等级制度对比，最后PK出胜利的一方。

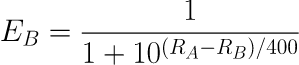


这条对比公式就是Elo Score等级分制度。Elo的最初为了提供国际象棋中，更好地对不同的选手进行等级划分。在现在很多的竞技运动或者游戏中都会采取Elo等级分制度对选手或玩家进行等级划分，如足球、篮球、棒球比赛或LOL，DOTA等游戏。

在这里我们将基于国际象棋比赛，大致地介绍下Elo等级划分制度。在上图中Eduardo在窗户上写下的公式就是根据Logistic Distribution计算PK双方（A和B）对各自的胜率期望值计算公式。假设A和B的当前等级分为R\_A*R*​*A*​​何R\_B*R*​*B*​​，则A对B的胜率期望值为：



B对A的胜率期望值为:



如果棋手A在比赛中的真实得分S\_A*S*​*A*​​（胜1分，和0.5分，负0分）和他的胜率期望值E\_A*E*​*A*​​不同，则他的等级分要根据以下公式进行调整：



在国际象棋中，根据等级分的不同K值也会做相应的调整：

* ≥2400，K=16
* 2100~2400分，K=24
* ≤2100，K=32

因此我们将会用以表示某场比赛数据的特征向量为（加入A与B队比赛）：[A队Elo score, A队的T,O和M表统计数据，B队Elo score, B队的T,O和M表统计数据]

## **四、基于数据进行模型训练和预测**

### **4.1 开发环境准备**

我们将会使用到python的pandas，numpy，scipy和sklearn库，不过实验楼中已经安装了numpy，所以在实验前，我们需要先利用pip命令安装另外三个Python库。

$ pip install pandas

$ pip install scipy

$ pip install sklearn

$ pip install cookiecutter

在安装完所需的实验库之后，初始化工程

cookiecutter https://github.com/audreyr/cookiecutter-pypackage.git

根据提示输入相关信息，我们把工程命名为nba\_prediction。然后将我们下载的数据文件保存到nba\_prediction/data下面。初始化完成后的工程目录如下：

├── AUTHORS.rst

├── CONTRIBUTING.rst

├── HISTORY.rst

├── LICENSE

├── MANIFEST.in

├── Makefile

├── README.rst

├── docs

│ ├── Makefile

│ ├── authors.rst

│ ├── conf.py

│ ├── contributing.rst

│ ├── history.rst

│ ├── index.rst

│ ├── installation.rst

│ ├── make.bat

│ ├── readme.rst

│ └── usage.rst

├── nba\_prediction

│ ├── \_\_init\_\_.py

│ ├── cli.py

│ ├── data

│ │ ├── 15-16Miscellaneous\_Stat.csv

│ │ ├── 15-16Opponent\_Per\_Game\_Stat.csv

│ │ ├── 15-16Team\_Per\_Game\_Stat.csv

│ │ └── 16-17Schedule.csv

│ └── nba\_prediction.py

├── requirements\_dev.txt

├── setup.cfg

├── setup.py

├── tests

│ ├── \_\_init\_\_.py

│ └── test\_nba\_prediction.py

├── tox.ini

└── travis\_pypi\_setup.py

在data文件夹中，包含了2015~2016年的NBA数据T,O和M表，及经处理后的常规赛和挑战赛的比赛数据2015~16result.csv，这个数据文件是我们通过在basketball-reference.com的2015-16 Schedule and result的几个月份比赛数据中提取得到的，其中包括三个字段：

* WTeam: 比赛胜利队伍
* LTeam: 失败队伍
* WLoc: 胜利队伍一方所在的为主场或是客场 另外一个文件就是16-17Schedule.csv，也是经过我们加工处理得到的NBA在2016~2017年的常规赛的比赛安排，其中包括两个字段：
* Vteam: 访问/客场作战队伍
* Hteam: 主场作战队伍

### **4.2 代码实现**

建具体工程代码

## **五、总结**

我们利用Basketball-reference.com的部分统计数据，计算每支nba比赛队伍的Elo socre，和利用这些基本统计数据评价每支队伍过去的比赛情况，并且根据国际等级划分方法Elo Score对队伍现在的战斗等级进行评分，最终结合这些不同队伍的特征判断在一场比赛中，哪支队伍能够占到优势。但在我们的预测结果中，我们没有给出绝对的正负之分，而是给出胜算较大一方的队伍能够赢另外一方的概率。当然在这里，我们所采用评价一支队伍性能的数据量还太少（只采用了15~16年一年的数据），如果想要更加准确、系统的判断，有兴趣的你当然可以从各种统计数据网站中获取到更多年份，更加全面的数据。结合不同的回归、决策机器学习模型，搭建一个更加全面，预测准确率更高的模型。在[kaggle](https://www.kaggle.com/)中有相关的篮球预测比赛项目，有兴趣的同学可尝试一下。

## **六、参考阅读**

* [知乎：在哪能看到最全面细致的NBA数据统计](https://www.zhihu.com/question/19952290)
* [How I won my NCAA tournament bracket pool using machine learning](http://ftw.usatoday.com/2016/04/villanova-bracket-winner)

## **七、思考**

我们只是利用了scikit-learn提供的Logisitc Regression方法进行回归模型的训练，你可否尝试scikit-learn中的其他机器学习方法，或者其他类似于TensorFlow的开源框架，结合我们所提供的数据集进行训练。

