**橄榄球分析预测**

**【实验内容】**

现有两份橄榄球进攻与防守数据的数据文件offense.csv和defense.csv。

offense.csv文件的数据样例

[view plain](https://www.ipieuvre.com/e/563/311/9539) [copy](https://www.ipieuvre.com/e/563/311/9539)

1. Team,G,Pts/G,Yds/G,PasY/G,RusY/G,1stD/G,3rdM,3rdD%,4thM,4thD%,PEN,PYds,ToP/G
2. LA Rams,16,29.9,373.5,251.4,122.1,19.4,88,41.1,5,41.7,107,922,1.254166667
3. New England,16,28.6,406.8,288.7,118.1,24.3,82,40.6,8,61.5,95,835,1.275694444
4. Philadelphia,16,28.6,380.1,247.9,132.2,21.1,96,41.7,17,65.4,116,962,1.361805556
5. New Orleans,16,28,400.3,270.9,129.4,20.8,73,37.6,12,80,105,962,1.311111111

defense.csv文件的数据样例

[view plain](https://www.ipieuvre.com/e/563/311/9539) [copy](https://www.ipieuvre.com/e/563/311/9539)

1. Team,G,Pts/G,Yds/G,RushYds/G,PassYds/G,Int,IntTD,FF,DefTD,Total,PD,Sack
2. Minnesota,16,15.8,275.9,83.6,192.4,14,1,6,1,558,71,37
3. Jacksonville,16,16.8,286.1,116.3,169.9,21,2,17,7,633,81,55
4. LA Chargers,16,17,328.4,131.1,197.3,18,2,13,3,699,79,43
5. Philadelphia,16,18.4,306.5,79.2,227.3,19,1,13,5,599,92,38

分析预测橄榄球控球能力排名。

**【实验环境】**

Linux Ubuntu 16.04

Python3.6

PyCharm

**【实验步骤】**

1.在Linux上新建一个目录/data/NFL目录

[view plain](https://www.ipieuvre.com/e/563/311/9539) [copy](https://www.ipieuvre.com/e/563/311/9539)

1. mkdir -p /data/NFL

2.切换到/data/NFL目录下，使用wget 命令，从http://192.168.1.100:60000/allfiles/NFL/目录下，将实验内容里提及的数据文件offense.csv和defense.csv下载到/data/NFL目录下。

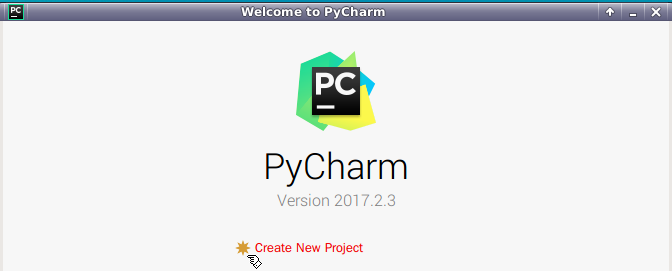
[view plain](https://www.ipieuvre.com/e/563/311/9539) [copy](https://www.ipieuvre.com/e/563/311/9539)

1. cd /data/NFL
2. wget http://192.168.1.100:60000/allfiles/NFL/offense.csv
3. wget http://192.168.1.100:60000/allfiles/NFL/defense.csv

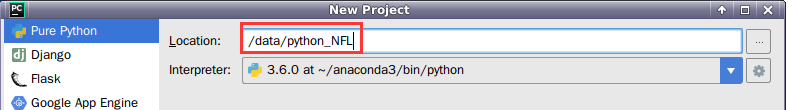
3.在图形化界面打开PyCharm，

[](https://www.ipieuvre.com/doc/exper/40150422-91ad-11e9-beeb-00215ec892f4/img/aa9eba89-27ff-4a21-a01d-1564acf46d2e.jpg)

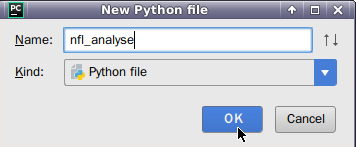
新建Python项目

[](https://www.ipieuvre.com/doc/exper/40150422-91ad-11e9-beeb-00215ec892f4/img/fe596691-e7e8-4d71-8899-9c6f37a8c4b9.png)

项目位置：/data/python\_NFL。

[](https://www.ipieuvre.com/doc/exper/40150422-91ad-11e9-beeb-00215ec892f4/img/39553b98-d86b-4925-afd4-a5e34f319084.png)

4.在python\_NFL项目下，新建Python File ，名为nfl\_analyse.

[](https://www.ipieuvre.com/doc/exper/40150422-91ad-11e9-beeb-00215ec892f4/img/3a3890b6-c970-45d6-9a46-1e4f32f4036f.png)

5.编写实验代码

导入实验所需的模块

[view plain](https://www.ipieuvre.com/e/563/311/9539) [copy](https://www.ipieuvre.com/e/563/311/9539)

1. **import** pandas as pd
2. **import** numpy as np
3. from matplotlib **import** pyplot as plt2
4. from sklearn.linear\_model **import** LinearRegression
5. from sklearn.model\_selection **import** train\_test\_split

导入要分析的数据

[view plain](https://www.ipieuvre.com/e/563/311/9539) [copy](https://www.ipieuvre.com/e/563/311/9539)

1. offense=pd.read\_csv("/data/NFL/offense.csv")
2. defense=pd.read\_csv("/data/NFL/defense.csv")

数据探索性分析，分析各个特征之间的相关性

[view plain](https://www.ipieuvre.com/e/563/311/9539) [copy](https://www.ipieuvre.com/e/563/311/9539)

1. #每场平均得分与平均进行码数的散点图
2. plt2.subplot2grid((2,2),(0,0))
3. plt2.scatter(combined.OffYPG,combined.OffPPG)
4. plt2.xlabel("Yards per Game")
5. plt2.ylabel("Points per Game")
6. plt2.title("Offense Yards vs Points per Game")
8. #平均实守码数和平均失守分值的散点图
9. plt2.subplot2grid((2,2),(0,1))
10. plt2.scatter(combined.DefYPG,combined.DefPPG)
11. plt2.xlabel("Yards Allowed per Game")
12. plt2.ylabel("Points Alloed per Game")
13. plt2.title("Defense Yards vs Points per Game")
15. #平均每场传球码数与每场平均得分散点图
16. plt2.subplot2grid((2,2),(1,0))
17. plt2.scatter(combined['OffPassYPG'],combined.OffPPG)
18. plt2.xlabel("average passing yards per game")
19. plt2.ylabel("Points per Game")
20. plt2.title("Time of Possession vs Points per Game")
22. #平均得分和球队拥有控球权时间的散点图
23. plt2.subplot2grid((2,2),(1,1))
24. plt2.scatter(combined['ToP/G'],combined.OffPPG)
25. plt2.xlabel("Time of Possession(Seconds)")
26. plt2.ylabel("Points per Game")
27. plt2.title("Time of Possession vs Points per Game")
28. print(combined.corr())

数据变换，构建度量攻防能力的指标

[view plain](https://www.ipieuvre.com/e/563/311/9539) [copy](https://www.ipieuvre.com/e/563/311/9539)

1. #计算球队的进攻传球能力分值
2. combined['OPassStrength']=max(combined.OffPassYPG)-combined.OffPassYPG
3. combined['OPassStrength']=(1-combined.OPassStrength/max(combined.OPassStrength))\*\*100
4. #计算球队的进攻冲击能力分值
5. combined['ORushStrength']=max(combined.OffRushYPG)-combined.OffRushYPG
6. combined['ORushStrength']=(1-combined.ORushStrength/max(combined.ORushStrength))\*\*100
7. #计算球队的进攻得分分值
8. combined['OPPGStrength']=max(combined.OffPPG)-combined.OffPPG
9. combined['OPPGStrength']=(1-combined.OPPGStrength/max(combined.OPPGStrength))\*\*100
10. #计算球队的进攻码数分值
11. combined['OYPGStrength']=max(combined.OffYPG)-combined.OffYPG
12. combined['OYPGStrength']=(1-combined.OYPGStrength/max(combined.OYPGStrength))\*\*100
13. #将变量综合成用户的攻击能力
14. combined['OffStrength']=(combined.OPassStrength+combined.ORushStrength+combined.OPPGStrength+combined.OYPGStrength)/4
15. #计算球对截断对手传球能力指标
16. combined['DPassStrength']=max(combined.DefPassYPG)-combined.DefPassYPG
17. combined['DPassStrength']=combined.DPassStrength/max(combined.DPassStrength)\*\*100
18. #计算球队防守对手冲球能力分值
19. combined['DRushStrength']=max(combined.DefRushYPG)-combined.DefRushYPG
20. combined['DRushStrength']=combined.DRushStrength/max(combined.DRushStrength)\*\*100
21. #计算球赛失防分值
22. combined['DPPGStrength']=max(combined.DefPPG )-combined.DefPPG
23. combined['DPPGStrength']=combined.DPPGStrength/max(combined.DPPGStrength)\*\*100
24. #计算球队总失防码数分值
25. combined['DYPGStrength']=max(combined.DefYPG)-combined.DefYPG
26. combined['DYPGStrength']=combined.DYPGStrength/max(combined.DYPGStrength)\*\*100
27. combined['DefStrength']=(combined.DPassStrength+combined.DRushStrength+combined.DPPGStrength+combined.DYPGStrength)/4
28. combined['ODStrength']=combined['OffStrength']+combined['DefStrength']
29. print(combined.columns)

注意：在计算球队防守能力指标中，我们并没有用1减去各个球队和最大值之间的比值，因为对于防守，比较小的数值表明球队有较强的防守能力，而不是如攻击能力计算中分值较大代表球队强大的进攻能力。

数据预处理，随机拆分30%为测试数据，70%数据为训练数据

[view plain](https://www.ipieuvre.com/e/563/311/9539) [copy](https://www.ipieuvre.com/e/563/311/9539)

1. train,test=train\_test\_split(combined,test\_size=0.3)
2. train\_x=train.iloc[:,-6:-5]
3. print(train\_x)
4. train\_y=train.iloc[:,13]
5. test\_x=test.iloc[:,-6:-5]
6. test\_y=test.iloc[:,13]

建立模型，并用测试数据进行预测

[view plain](https://www.ipieuvre.com/e/563/311/9539) [copy](https://www.ipieuvre.com/e/563/311/9539)

1. model=LinearRegression()
2. model.fit(train\_x,train\_y)
3. test\_pred=model.predict(test\_x)

模型评估，输出模型的残差平方和与方差分数

[view plain](https://www.ipieuvre.com/e/563/311/9539) [copy](https://www.ipieuvre.com/e/563/311/9539)

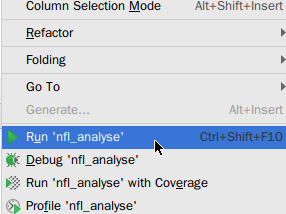
1. print("true value:\n",test\_y)
2. print("pred value:\n",test\_pred)
3. print('Residual sum of squares: %.2f' % np.mean(test\_y-test\_pred)\*\*2 )
4. print('variance score: %.2f' % model.score(test\_x,test\_y))

6.完整代码：

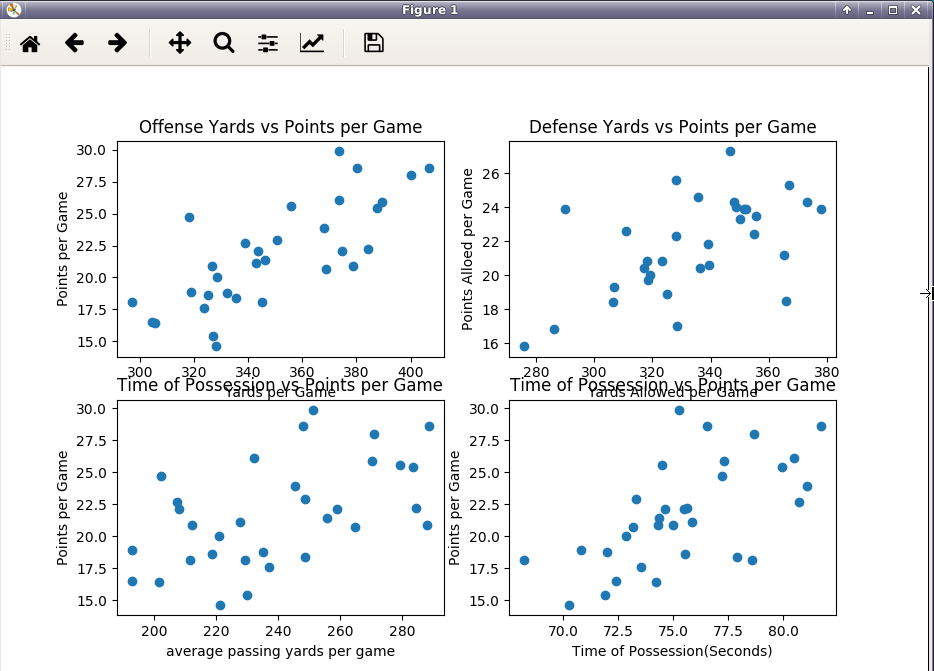
[view plain](https://www.ipieuvre.com/e/563/311/9539) [copy](https://www.ipieuvre.com/e/563/311/9539)

1. **import** pandas as pd
2. **import** numpy as np
3. from matplotlib **import** pyplot as plt2
4. from sklearn.linear\_model **import** LinearRegression
5. from sklearn.model\_selection **import** train\_test\_split
6. offense=pd.read\_csv("/data/NFL/offense.csv")
7. defense=pd.read\_csv("/data/NFL/defense.csv")
8. offense['ToP/G']=offense['ToP/G']\*60
9. combined=pd.merge(offense,defense,how="outer",on='Team')
10. combined.rename(columns={'G\_x':"Games","Pts/G\_x":"OffPPG","Yds/G\_x":"OffYPG","PasY/G":"OffPassYPG","RusY/G":"OffRushYPG","Pts/G\_y":"DefPPG","Yds/G\_y":"DefYPG","RushYds/G":"DefRushYPG","PassYds/G":"DefPassYPG"},inplace=True)
11. print(combined.corr())
13. plt2.subplot2grid((2,2),(0,0))
14. plt2.scatter(combined.OffYPG,combined.OffPPG)
15. plt2.xlabel("Yards per Game")
16. plt2.ylabel("Points per Game")
17. plt2.title("Offense Yards vs Points per Game")
18. plt2.subplot2grid((2,2),(0,1))
19. plt2.scatter(combined.DefYPG,combined.DefPPG)
20. plt2.xlabel("Yards Allowed per Game")
21. plt2.ylabel("Points Alloed per Game")
22. plt2.title("Defense Yards vs Points per Game")
23. plt2.subplot2grid((2,2),(1,0))
24. plt2.scatter(combined['OffPassYPG'],combined.OffPPG)
25. plt2.xlabel("average passing yards per game")
26. plt2.ylabel("Points per Game")
27. plt2.title("Time of Possession vs Points per Game")
28. plt2.subplot2grid((2,2),(1,1))
29. plt2.scatter(combined['ToP/G'],combined.OffPPG)
30. plt2.xlabel("Time of Possession(Seconds)")
31. plt2.ylabel("Points per Game")
32. plt2.title("Time of Possession vs Points per Game")
34. combined['OPassStrength']=max(combined.OffPassYPG)-combined.OffPassYPG
35. combined['OPassStrength']=(1-combined.OPassStrength/max(combined.OPassStrength))\*\*100
36. combined['ORushStrength']=max(combined.OffRushYPG)-combined.OffRushYPG
37. combined['ORushStrength']=(1-combined.ORushStrength/max(combined.ORushStrength))\*\*100
38. combined['OPPGStrength']=max(combined.OffPPG)-combined.OffPPG
39. combined['OPPGStrength']=(1-combined.OPPGStrength/max(combined.OPPGStrength))\*\*100
40. combined['OYPGStrength']=max(combined.OffYPG)-combined.OffYPG
41. combined['OYPGStrength']=(1-combined.OYPGStrength/max(combined.OYPGStrength))\*\*100
42. combined['OffStrength']=(combined.OPassStrength+combined.ORushStrength+combined.OPPGStrength+combined.OYPGStrength)/4
43. combined['DPassStrength']=max(combined.DefPassYPG)-combined.DefPassYPG
44. combined['DPassStrength']=combined.DPassStrength/max(combined.DPassStrength)\*\*100
45. combined['DRushStrength']=max(combined.DefRushYPG)-combined.DefRushYPG
46. combined['DRushStrength']=combined.DRushStrength/max(combined.DRushStrength)\*\*100
47. combined['DPPGStrength']=max(combined.DefPPG )-combined.DefPPG
48. combined['DPPGStrength']=combined.DPPGStrength/max(combined.DPPGStrength)\*\*100
49. combined['DYPGStrength']=max(combined.DefYPG)-combined.DefYPG
50. combined['DYPGStrength']=combined.DYPGStrength/max(combined.DYPGStrength)\*\*100
51. combined['DefStrength']=(combined.DPassStrength+combined.DRushStrength+combined.DPPGStrength+combined.DYPGStrength)/4
52. combined['ODStrength']=combined['OffStrength']+combined['DefStrength']
53. train,test=train\_test\_split(combined,test\_size=0.3)
54. train\_x=train.iloc[:,-6:-5]
55. train\_y=train.iloc[:,13]
56. test\_x=test.iloc[:,-6:-5]
57. test\_y=test.iloc[:,13]
59. model=LinearRegression()
60. model.fit(train\_x,train\_y)
61. test\_pred=model.predict(test\_x)
62. print("true value:\n",test\_y)
63. print("pred value:\n",test\_pred)
64. print('Residual sum of squares: %.2f' % np.mean(test\_y-test\_pred)\*\*2 )
65. print('variance score: %.2f' % model.score(test\_x,test\_y))
66. plt2.show()

7.在nfl\_analyse文件中，右键点击Run“nfl\_analyse”，运行实验代码

[](https://www.ipieuvre.com/doc/exper/40150422-91ad-11e9-beeb-00215ec892f4/img/f9eee4cf-7466-4d63-9853-b10cf0a4bb01.png)

8.运行结果如下：

[](https://www.ipieuvre.com/doc/exper/40150422-91ad-11e9-beeb-00215ec892f4/img/0dd7625d-a9ba-4518-a90a-3b0b10db8c90.png)

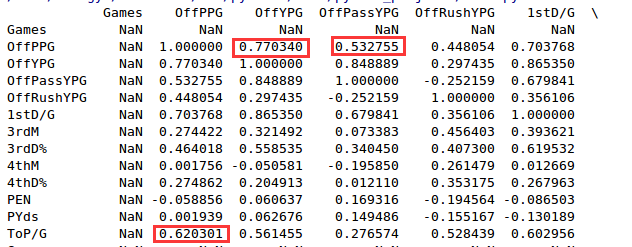
由图中的第一个散点图可知，平均进攻码与平均进攻得分呈正相关，当平均进攻码数增加时，平均进攻得分也逐渐上升。

由图中的第二个散点图可知，球对的平均失守码数与平均失守分值之间也存在一些正相关，但是没有两个得分变量之间的相关性强烈。

由图中的第三个散点图可知，平均每场传球码数与每场平均得分之间也存在一些正相关，但是没有两个得分变量之间的相关性强烈。

由图中的第四个散点图可知，球对的平均得分数与球队拥有的控球时间之间也存在一些正相关，但是没有两个得分变量之间的相关性强烈。

通过各个特征的系数相关性表也可以看出。

[](https://www.ipieuvre.com/doc/exper/40150422-91ad-11e9-beeb-00215ec892f4/img/d26c2cad-3383-49a0-8e53-1c9c961852e4.png)

模型评估时，可以看出残差平方和与方差得分都很小，模型较好。

[](https://www.ipieuvre.com/doc/exper/40150422-91ad-11e9-beeb-00215ec892f4/img/980609c5-451d-4491-a8d5-ecaef7c53f6b.png)