实验三

PB19000196 晏瑞然

实验要求：

给定一个数据集和预测任务，需要分析数据以及抽取特征。

实验过程：

导入模块：

主要导入三个python数据分析重要的库：numpy, pandas, matplotlib.

读取数据：

用pandans库中read_csv读取实验所给的exp3.csv文件，将文件中的数据存储成pandas库中的dataframe的格式。打印前5条数据观察数据大致的格式如下：

```
       queueId  mapId  seasonId  team1_win  team1_firstBlood  \
index
0          420     11        13          0              True
1          420     11        13          1             False
2          420     11        13          0             False
3          420     11        13          0              True
4          420     11        13          1              True

       team1_firstTower  team1_firstInhibitor  team1_firstBaron  \
index
0                 False                 False             False
1                  True                  True             False
2                  True                 False             False
3                 False                 False             False
4                  True                  True             False

       team1_firstDragon  team1_firstRiftHerald  ...  player9_goldEarned  \
index                                            ...
0                  False                  False  ...                6967
1                   True                   True  ...                8133
2                   True                   True  ...               11868
3                   True                  False  ...                6703
4                  False                   True  ...                6823

       player9_role  player9_lane  player10_championId  player10_kills  \
index
0       DUO_SUPPORT          NONE                  157              3
1              SOLO        MIDDLE                  555              3
2         DUO_CARRY        BOTTOM                  122              2
3       DUO_SUPPORT        BOTTOM                   23              6
4               DUO        MIDDLE                   25              5

      player10_deaths player10_assists  player10_goldEarned  player10_role  \
index
0                   0                8                8168            DUO
1                   7                4                6943    DUO_SUPPORT
2                   2                1                9685           SOLO
3                   2               10               13109           SOLO
4                   7                8                6809    DUO_SUPPORT

       player10_lane
index
0               NONE
1             BOTTOM
2                TOP
3                TOP
4             BOTTOM

[5 rows x 80 columns]
```

可以大致看到各个特征与数据的结构，再打印出总表的大小，得到其表规格为80000×80，即80000条数据，每条数据80个特征。

数据分析与特征抽取：

1.比赛特征

主要工作： 分析每次比赛的地图、队列、赛季特征。

首先，查看mapId,queueId,seasonId的数据分布，可以看出mapId有两种11和12，其中11个数有70001次，12有999次。 queueId有7中，各自频率如下。 seasonId只有一种取值13,故该特征可直接去除。

```
11    79001
12      999
Name: mapId, dtype: int64
420     50146
430     20966
440      4016
700      2594
900      1023
450       999
1020      256
Name: queueId, dtype: int64
13    80000
Name: seasonId, dtype: int64
```

同时发现有map为12的个数与queue为450的数据个数相同，故猜测queue为450与map为12等价。 下面进行验证：

```
print(all(data[data['mapId']==12]['queueId']==450))
del data['mapId']
```

```
True
```

结果与我们的猜想相同，故mapId也没有意义，可以去掉。

再来看看queueId与胜负的关系，得到如下crosstab：

team1_win	0	1
queueId		
420	25434	24712
430	11211	9755
440	2488	1528
450	498	501
700	1182	1412
900	517	506
1020	145	111

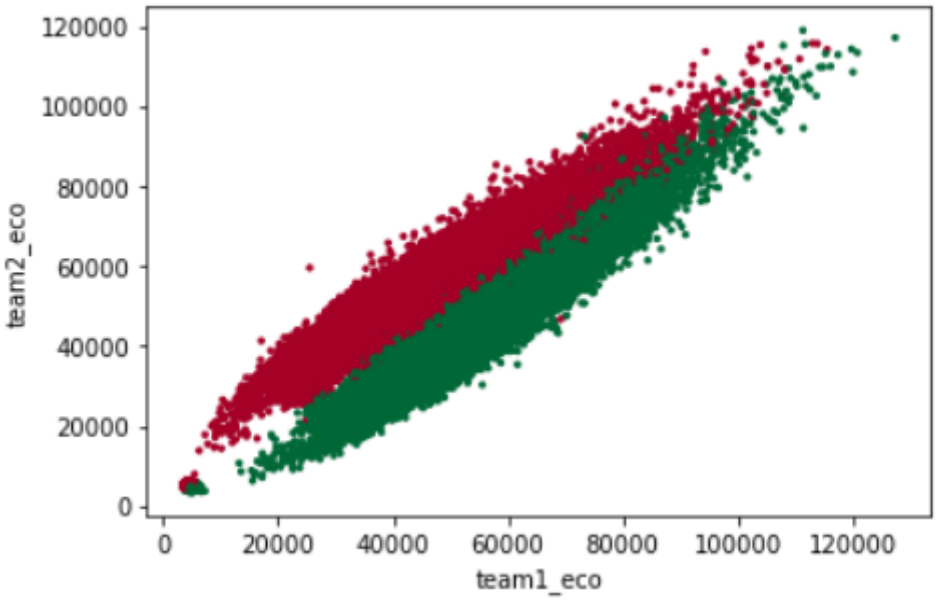
可以看出大多数情况的queueId都在420,430之间且基本胜率接近50%，故也可以认为其与胜负关系无关，可以删除。

2.队伍特征

主要任务： 通过每个队伍的每个选手的数据，组合成队伍的特征(整体经济、整体人头数)，如根据每个人的经济得到队伍经济。

2.1队伍经济分析

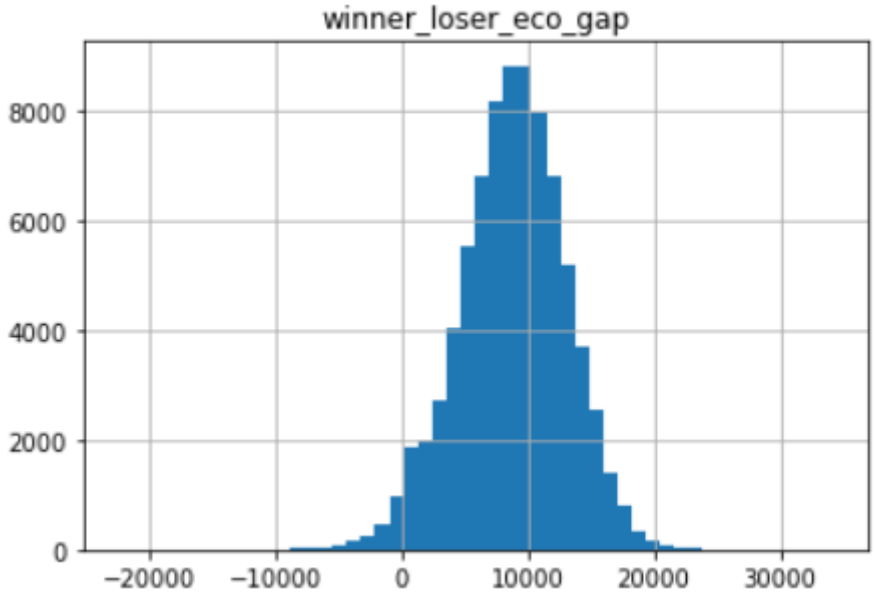
1. 获取每个队伍的经济及经济差，并将得到的数据加入到特征表中。画出经济散点图如下图所示，其中绿色代表team1赢，红色代表team2赢。



2. 画出经济差正负与胜负关系的列联表，其中eco_gap=True代表经济差(team1-team2)大于0,eco_gap=False反之。根据列联表检验，可以认为游戏胜负与经济是否领先相关。

eco_gap	False	True
team1_win		
0	40573	902
1	981	37544

3. 得到胜方与负方经济差分布图及各个统计量，可以看出胜负方经济差大致符合一个正态分布，其均值在8763左右。



	winner_loser_eco_gap
count	80000.000000
mean	8763.360787
std	4207.051713
min	-22459.000000
25%	6110.000000
50%	8914.000000
75%	11610.000000
max	34036.000000

结论：

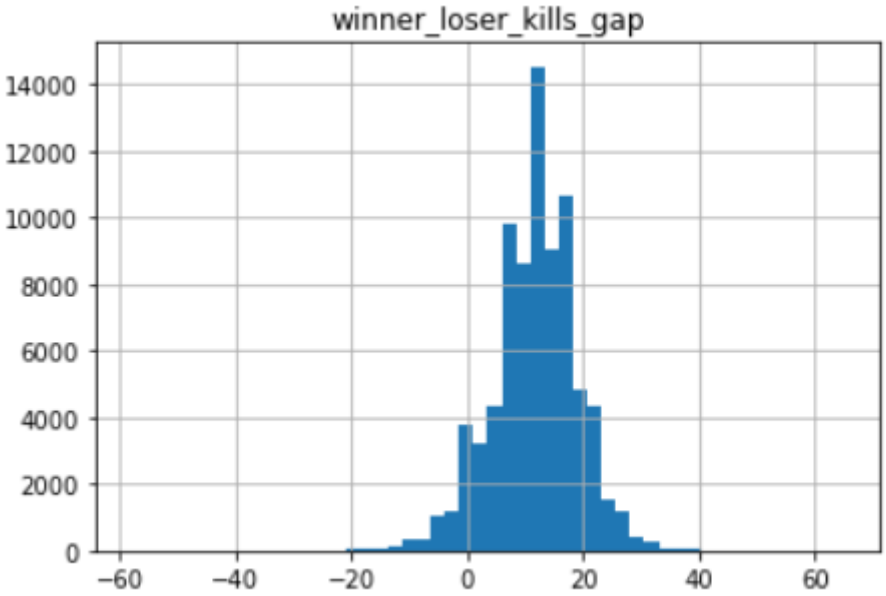
- 经济是否领先与胜负相关，且经济高的一方更容易获胜，这也很符合常理。
- 游戏结束时的胜负方经济差大概符合一个均值为8763的正态分布，其各统计量如上图所示。

2.2队伍杀敌数分析

1. 得到每个队伍的杀敌数及杀敌数差，将其加入特征表。因杀敌数相对经济更加集中所以没必要画散点图。
2. 画出人头差正负与胜负关系的列联表，其中kills_gap=True代表经济差(team1-team2)大于0,kills_gap=False反之。根据列联表检验，可以认为游戏胜负与经济是否领先相关。

kills_gap	False	True
team1_win		
0	39327	2148
1	2626	35899

3. 得到胜负各方的杀敌数分布图及各个统计量，可以看出胜负方人头差大致符合一个正态分布，其均值在12左右。



	winner_loser_kills_gap
count	80000.000000
mean	11.714688
std	7.353117
min	-58.000000
25%	7.000000
50%	12.000000
75%	16.000000
max	65.000000

结论：

- 人头数是否领先与胜负相关，且经济高的一方更容易获胜，这也很符合常理。
- 游戏结束时的胜负方人头差大概符合一个均值为12的正态分布，其各统计量如上图所示。

3. 地图资源特征

主要任务： 分析各个地图资源(如1塔，1龙等)特征对胜负的影响，用曼哈顿距离度量两者距离(相关度)，曼哈顿距离即为team1拿了该资源却输了和未拿该资源却赢了的比赛总和。

3.0 数据预处理

进行分析前首先进行数据类型转换，将TRUE，FALSE字符类型转换成0，1的int型。

3.1 一血

查看是否拿一血与胜负的关系列联表：

team1_firstBlood	0	1
team1_win		
0	24265	17210
1	15162	23363

计算曼哈顿距离及拿了该资源后的胜率：

```
Manhattan distance is 32372
win rate is 0.5758262884184063
```

3.2 一塔

查看是否拿一塔与胜负的关系列联表：

team1_firstTower	0	1
team1_win		
0	29646	11829
1	10864	27661

计算曼哈顿距离及拿了该资源后的胜率：

```
Manhattan distance is 22693
win rate is 0.7004558115978728
```

3.3 一先锋

查看是否拿一先锋与胜负的关系列联表：

team1_firstInhibitor	0	1
team1_win		
0	38626	2849
1	11189	27336

计算曼哈顿距离及拿了该资源后的胜率：

```
Manhattan distance is 14038
win rate is 0.9056153718734471
```

3.4 一男爵

查看是否拿一男爵与胜负的关系列联表：

team1_firstBaron	0	1
team1_win		
0	37497	3978
1	20144	18381

计算曼哈顿距离及拿了该资源后的胜率：

```
Manhattan distance is 24122
win rate is 0.8220850664162083
```

3.5 一龙

查看是否拿一龙与胜负的关系列联表：

team1_firstDragon	0	1
team1_win		
0	27024	14451
1	15266	23259

计算曼哈顿距离及拿了该资源后的胜率：

```
Manhattan distance is 29717
win rate is 0.6167859984089101
```

3.6 一水晶

查看是否拿一水晶与胜负的关系列联表：

team1_firstRiftHerald	0	1
team1_win		
0	26728	14747
1	16460	22065

计算曼哈顿距离及拿了该资源后的胜率：

```
Manhattan distance is 31207
win rate is 0.5993969357818103
```

分析：

可以看出拿了地图资源胜率会提高，这在实际中也是显然的。同时也可以发现这些数据中一些有趣的性质。
首先一血、一塔的矩阵正反对角线上的数基本相同，但其他资源的对应矩阵明显有左上大于右下，左下大于右上的性质。
难道team1和team2拿这些资源的胜率有偏差？显然不是，经分析，出现该种状态的原因是**有很多比赛两边都没拿该资源就结束了！！！**
所以左侧数据大于右侧。这时计算拿了该资源后的胜率就比较有意义，现实中lol比赛分析也都如此，会分析拿了某项资源后的胜率。
数据中也有一些反直觉的地方，比如拿了一先锋后的胜率比一大龙的胜率还高，理论上越偏向后期的资源应该对胜率影响越大，**推测比赛数据可能有一定偏差，取的是结束时间较早的数据**，所以先锋作为前期的重要资源对胜负影响就很大，而两边可能都没打大龙游戏就结束了，所以大龙对胜负的影响就较小。

最后，根据曼哈顿距离分析，资源与胜负相关度从高到底排序为(距离近的相关度高)：一先锋 > 一塔 > 一男爵 > 一龙 > 一水晶 > 一血。

4.英雄及阵容特征

主要任务： 分析各英雄与阵容数据，如英雄胜率、英雄常见位置、阵容胜率等。

4.0 数据预处理

由于只分析英雄相关数据，故把所有玩家有关英雄的数据合在一张表中，得到championData。

	win/lose	championId	kills	deaths	assists	goldEarned	role	lane
0	0	107	2	4	0	5385	DUO_SUPPORT	NONE
1	1	76	10	5	6	11865	NONE	JUNGLE
2	0	51	7	6	2	12985	DUO_CARRY	BOTTOM
3	0	28	5	3	5	8953	NONE	JUNGLE
4	1	79	4	2	10	8347	NONE	JUNGLE
...	...	...	...	...	...	...	...	...
799995	1	235	3	5	13	8540	DUO_SUPPORT	BOTTOM
799996	0	157	0	7	4	8248	SOLO	MIDDLE
799997	1	7	10	2	13	11401	DUO	MIDDLE
799998	1	412	1	3	6	5607	DUO_SUPPORT	BOTTOM
799999	1	133	8	5	17	15290	NONE	JUNGLE

[800000 rows x 8 columns]

其中统计的是每场比赛中每个player使用的英雄的各项数据，每场比赛10个player，故有800000条数据。

4.1英雄胜率分析

生成英雄胜率表并找到其中的胜率最大最小值:

```
        1         2         3        4         5         6         7    \
0  0.494649  0.492625  0.511351  0.51063  0.494849  0.504712  0.471047

        8         9         10   ...       429       432       497       498  \
0  0.493365  0.507142  0.481537  ...  0.500113  0.529797  0.491146  0.474113

        516       517       518      523       555       875
0  0.490075  0.482556  0.476749  0.48902  0.482339  0.494461

[1 rows x 148 columns]
champion max win rate is 0.49262467806134397
champion min win rate is 0.49262467806134397
```

可以看出胜率最高为54%，最低为45%。 由此可见英雄对胜率的影响不算很大，在5%以内，该游戏还算比较平衡。

4.2 lane与role的分析

画出所有的lane与role的关系表:

lane	BOTTOM	JUNGLE	MIDDLE	NONE	TOP
role					
DUO	2168	0	26932	23408	13300
DUO_CARRY	117661	0	3753	18	1860
DUO_SUPPORT	118704	0	10358	109144	3875
NONE	0	142264	0	0	0
SOLO	9810	0	107897	0	108848

可以看出,role中所有的None的都来自打野选手，而lane中的none有各个角色。 故作出判断role中none是指打野选手并没有具体的角色而lane中的none是因为数据缺失造成的，要进行数据填补，而填补方式是用该英雄大的最多的lane来替换none。

接下来进行数据填补，同时得到每个英雄对应最常见lane的字典champion_lane_dict，再通过得到的字典，对原data表进行补全，补全过程及得到的字典见代码。

4.3 类型转换

将英雄，lane，role这些字符串类型特征转换为onehot编码，具体转换过程见代码，打印转换后的表格shape，可以看出已经转换完成。

```
print(data.shape)
```

```
(80000, 397)
```

4.4 得到英雄数据表

对各个英雄数据进行分析，算出登场率，场均人头，场均助攻等场均数据得到一张表，可以用该表衡量英雄各个方面的能力。

	1	2	3	4	5	\
Frequency	1682.000000	4271.000000	7136.000000	5456.000000	2718.000000	
win_rate	0.494649	0.492625	0.511351	0.510630	0.494849	
Kills	5.784780	5.741044	3.935959	4.507515	5.417954	
Assists	6.978597	5.959728	9.182315	7.805535	6.822296	
Deaths	5.174197	5.031843	4.659613	4.592559	5.527226	
Gold	10097.807372	10509.363381	9370.036155	11077.990103	9987.662987	

	6	7	8	9	\
Frequency	2653.000000	5388.000000	7988.000000	6231.000000	
win_rate	0.504712	0.471047	0.493365	0.507142	
Kills	5.703732	6.588901	5.268027	4.736158	
Assists	4.850358	6.097810	5.218703	8.776922	
Deaths	4.858651	4.466592	4.684402	5.149414	
Gold	11002.033924	10250.644581	10788.118553	9610.300433	

	10	...	429	432	497	\
Frequency	3358.000000	...	4437.000000	5655.000000	3840.000000	
win_rate	0.481537	...	0.500113	0.529797	0.491146	
Kills	4.618523	...	6.731575	2.403537	1.618490	
Assists	5.652174	...	5.996619	11.638550	12.442708	
Deaths	4.982430	...	5.630832	4.610610	4.735938	
Gold	10897.731388	...	11435.623845	7473.926437	7144.751302	

	498	516	517	518	523	\
Frequency	4172.000000	5038.000000	12784.000000	2387.000000	9290.000000	
win_rate	0.474113	0.490075	0.482556	0.476749	0.489020	
Kills	5.702301	3.170504	6.274406	4.265186	5.611087	
Assists	6.657958	7.645891	6.230288	7.492250	6.221313	
Deaths	5.011266	4.435093	5.565472	5.210306	5.320344	
Gold	11307.267018	9599.517864	10251.865535	9604.258065	11074.281485	

	555	875
Frequency	8295.000000	14171.000000
win_rate	0.482339	0.494461
Kills	6.184690	4.297297
Assists	7.194454	6.866417
Deaths	5.640145	5.239150
Gold	10184.585654	9577.649495

[6 rows x 148 columns]

数据存储：

将更改后的特征表存到exp3Data.csv中，得到的英雄数据表存到exp3ChampionData.csv中(见附带文件中的exp3ChampionData.csv文件)。

数据问题分析：

1. 有许多不合理的阵容存在，如有些比赛一个队有3个上路，这显然式不符合真实游戏情况的。
2. 有许多player_lane缺失的数据所有的player_role都为DUO_SUPPORT，这可能是获取的数据已经进行了填补，如再通过上述方式填补会有很多英雄的role都偏向DUO_SUPPORT，造成误差。
3. 如地图资源分析中所述，第一只先锋对游戏胜负影响最大这显然也不太符合常理，故游戏数据可能有一定偏差，如可能都是一些结束时间较早的数据。