

AI Academy 2019

2 место

**Экспорт признаков из JSON
в отдельные .csv файлы**



Генерация новых признаков



Обучение модели

Признаки из JSON

In [21]: 1 `print(*new_record.keys(), sep=', ')`

player_team, winner_team, duration, pre_game_duration, first_blood_time, first_blood_claimed, team_fight_participation, stuns, observer_wards_placed, sentry_wards_placed, creeps_stacked, camps_stacked, rune_pickups, tower_kills, roshan_kills, nearby_creep_death_count, damage_targets, radiant_tower_status, dire_tower_status, radiant_barracks_status, dire_barracks_status, hero_id, hero_pick_order, leaver_status, party_players, kills, deaths, assists, level, gold, net_worth, gold_spent, last_hits, denies, gold_per_min, xp_per_min, hero_damage, tower_damage, hero_healing, scaled_hero_damage, scaled_tower_damage, scaled_hero_healing, avg_kills_x16, avg_deaths_x16, avg_assists_x16, avg_gpm_x16, avg_xpm_x16, best_kills_x16, best_assists_x16, best_gpm_x16, best_xpm_x16, win_streak, best_win_streak, fight_score, farm_score, support_score, push_score, final_items, ability_upgrades, level_up_times, item_purchase_log, series, radiant_heroes, dire_heroes, wards, actions, pings, kills_log, deaths_log, gold_by_reason, id, skill

Герои в каждом матче

```
1 | new_record['dire_heroes'] + new_record['radiant_heroes']  
[84, 74, 4, 129, 73, 47, 6, 69, 8, 5]
```



	h1	h2	h3	h4	h5	h6	h7	h8	h9	h10	...	h109	h110	h111	h112	h113	h114	h119	h120	h121	h129
0																					
0	0	0	0	1	1	1	0	1	0	0	...	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	1	0	...	0	0	0	0	1	0	0	1	0	0
3	0	0	0	0	0	1	0	0	1	0	...	0	0	0	0	0	0	0	0	1	0
7	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1

5 rows × 117 columns

Предметы игрока

```
1 | new_record['final_items'] |
```

```
[249, 263, 40, 75, 63, 75, 0, 0, 0]
```



	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	...	i1022	i1023	i1024	i1025	i1026	i1027	i1028	i1029	i1030	i1032
0																					
12	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
16	1	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 234 columns

Способности игрока

```
1 new_record['ability_upgrades'][:10].
```

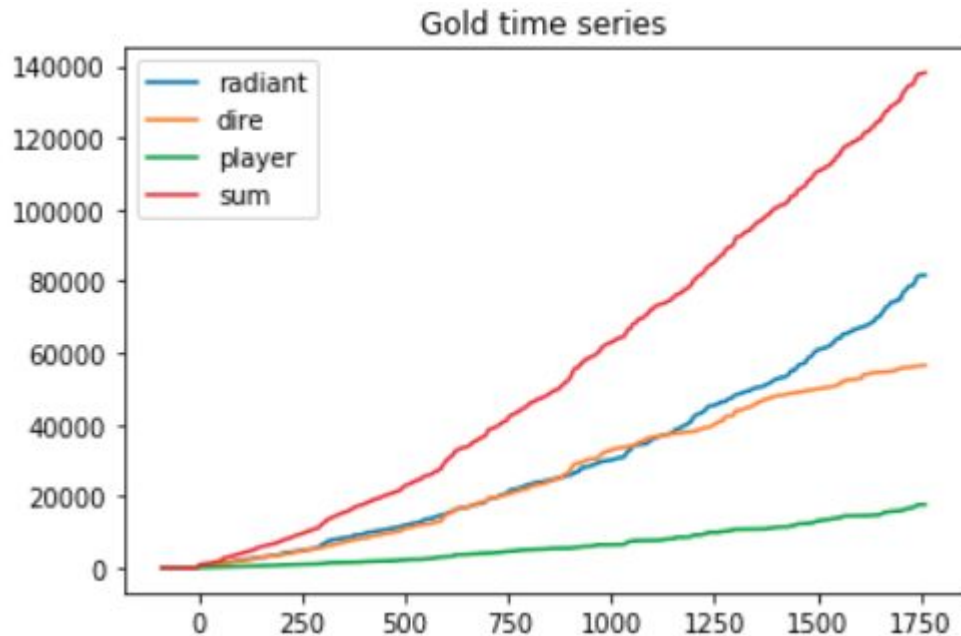
```
[5021, 5019, 5021, 5019, 5021, 5022, 5021, 5632, 5019, 5918]
```



	a5027	a5028	a5029	a5030	a5082	a5083	a5084	a5085	a5086	a5087
0										
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	4	4	4	3	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

10 rows × 1611 columns

Характеристики временных рядов



- Золото в конце игры
- Среднее значение
- Медианное значение
- Дисперсия ряда
- Среднее квадратическое

Новые признаки из JSON на финале

Gold by reasons - вектор полученного золота по каждой из возможных причин

Информация о действиях игрока:

Ping types - вектор числа сообщений команде по каждому из типов

Order types - вектор числа действий (кликов) по каждому из типов

```
1 gold_reasons_dfs[0].head()
```

	6	12	2	15	1	14	13	5	0	11
0	0	3754	0	0	-909	459	6771	0	1119	2867
1	1110	3571	0	0	-1105	0	9546	0	2113	2029
2	0	2020	0	0	-881	150	4833	0	1242	2854
3	1217	5062	0	0	-586	0	4865	0	2359	2821
7	225	3891	-558	0	-1192	0	1021	0	807	320

```
ord_1 ord_2 ord_3 ord_4 ord_5 ord_6 ord_7 ord_8 ord_9 ord_10 .
```

0	ord_1	ord_2	ord_3	ord_4	ord_5	ord_6	ord_7	ord_8	ord_9	ord_10	.
0	3170	0	0	713	2	227	4	7	0	116	
1	2608	0	0	219	34	2	2	25	0	4	
2	2939	124	27	528	20	1	5	4	0	51	
3	3278	4	42	817	23	6	4	27	0	85	
7	1985	13	11	497	7	54	4	5	2	36	

Генерация новых признаков

$KDA = (kills + assists) / deaths$

$KD = kills / deaths$

Gold ratio - отношение золота, заработанного игроком, к золоту команд.

Features per minute - отношение признаков к длительности матча

```
1 # features per minute
2 def pm(X, data_type):
3     col_to_pm = ['kills', 'deaths', 'assists',
4                 'denies', 'level', 'net_worth', 'gold', 'gold_spent', 'last_hits',
5                 'gold_per_min', 'xp_per_min', 'hero_damage', 'tower_damage',
6                 'hero_healing', 'scaled_hero_damage', 'scaled_tower_damage',
7                 'scaled_hero_healing', 'stuns', 'team_fight_participation',
8                 'observer_wards_placed', 'sentry_wards_placed', 'creeps_stacked',
9                 'camps_stacked', 'rune_pickups', 'tower_kills', 'roshan_kills',
10                'nearby_creep_death_count', 'fight_score',
11                'farm_score', 'support_score', 'push_score', 'radiant_gold', 'dire_gold', 'sum_gold', 'gxp_per_min_1', 'ka', 'kda']
12     for name in col_to_pm:
13         X[f'{name}_per_min'] = X[name]/X['duration']
14     print(data_type)
15     return X
```

Итоговый shape - **2560** столбцов

Обучение модели

Итоговый сабмит - среднее двух моделей:

CatBoost (глубина 9, 1000 итераций, GPU*)

LightGBM (глубина 10, 3000 итераций)

Итоговое качество: 0.91250 (ROC-AUC)



*Google Colaboratory предоставляет Titan X

Контакты



@lejabque



vk.com/lejabque



github.com/lejabque/ai_academy_2019