Выделенные характеристики

- gamescount - число сыгранных партий
- last_game_win - результат последней игры
- reg_day_type - день регистрации (ПН, ВТ ..)
- round_type_1 - число игр на 1 этапе чемпионата
- round_type_2 - число игр на 2 этапе чемпионата
- round_type_3 - число игр на 3 этапе чемпионата
- total_cards_done - число сыгранных карт
- total_length - общая время в игре
- total_lost - общее число сброшенных карт
- total_magic_used - общее число потраченной магии
- total_win - общее число побед
- turnir_counts - число сыгранных турниров
- turnir_wins - число побед в турнирах
- visits - число заходов в игру
- type_1 - число игр type 1
- type_2 - число игр type 2
- type_3 - число игр type 3
- type_4 - число игр type 4
- magic_for_card - среднее число магии потраченной на 1 карту
- time_for_card - среднее время выбора карты
- skill - уровень игрока
- time_for_game - среднее время на партию
- card_for_game - среднее число карт в игре
- magic_for_game - среднее число магии на карту
- games_for_visit - среднее число сыгранных игр за заход

In [182]:

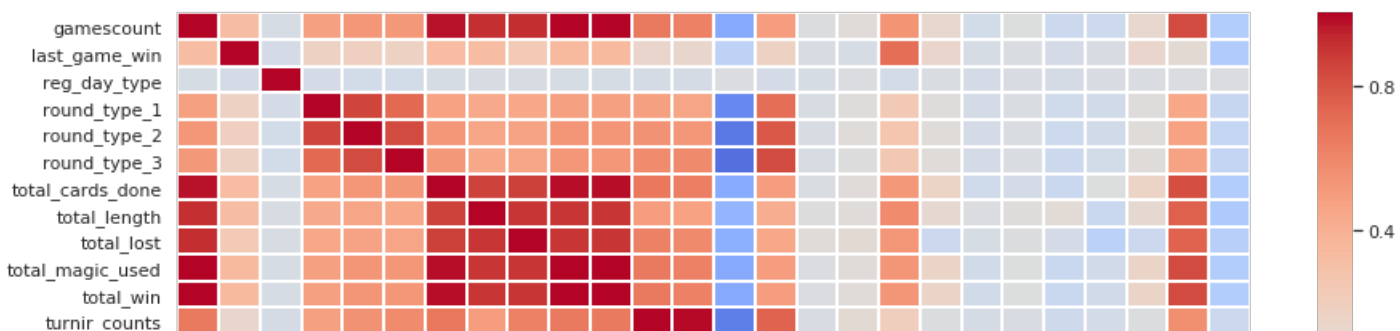
```
```

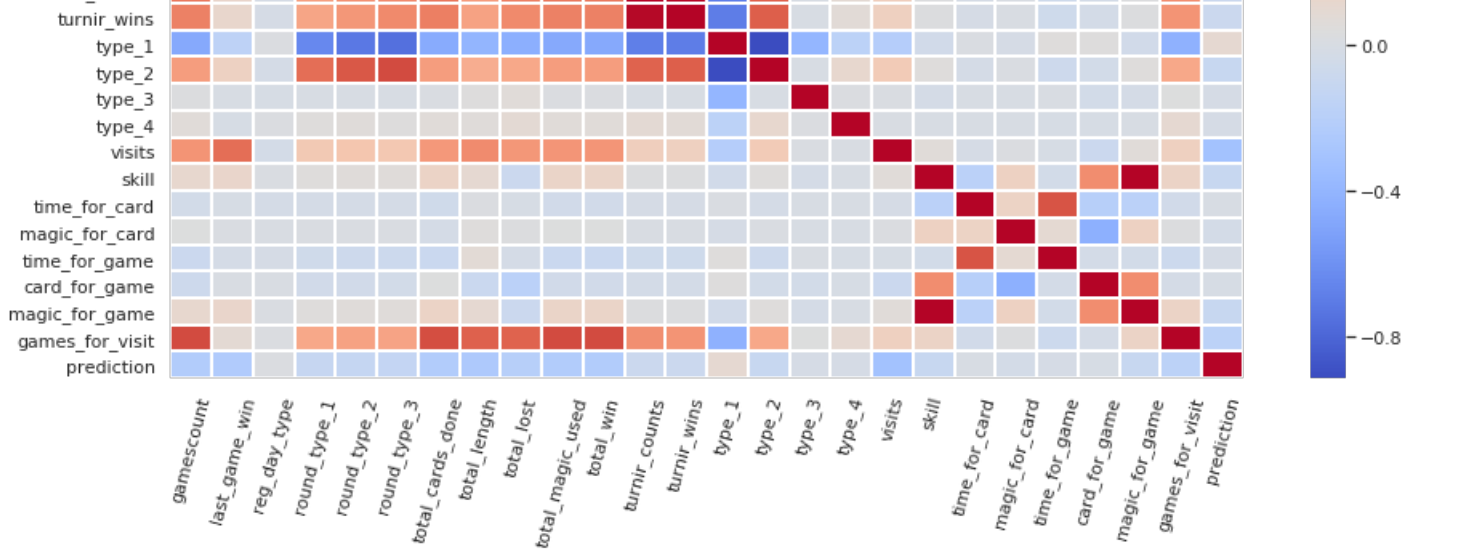
Люди регестрируются равномерно, день регистрации не на что не влияет

EAT

In [183]:

```
```





Оценим признаки по данным из heatmap

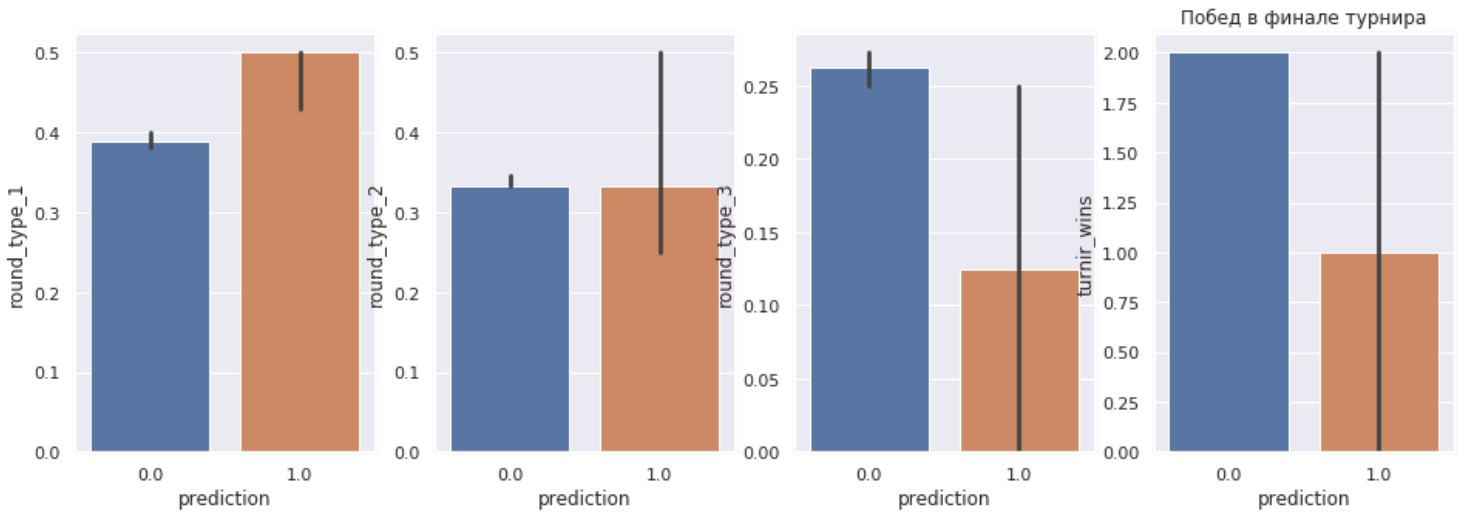
Больше всего нас интересует последняя строчка: `prediction`, сильнее всего НЕкосмотнавтность пользователя связана с числом дней с момента регистрации до последнего матча (проверенно во время тестов), чуть менее с числом активных дней в игре (проверенно во время тестов) и общим числом сыгранных карт / потраченной маны и времени, а также с рспределением по раундам в чемпионатах. Вопреки ожиданиям, космонавтность плохо коррелирует с количеством побед и уровнем игрока. По правилам игры дают много бонусов за сброс сетов и эти бонусы вляют на место в общей турнирной таблице. Есть вероятность что космонавтность связана со стратегией игрока, направленной на сброс максимально большого числа карт ради набора бонусов, и люди приходят к этому по мере получения опыта в игре. В пользу этой теории также гворит и то, что люди, играющие чаще других первый раунд обладают повышенной космонавтностью

- Судя по кореляции признаков космнавтность это негативное качество, как марсианин в нардах и более правильная тактика игры у тех кто **НЕ космонавт**
- `type_3` и `type4` вообще ни с чем не корелируют и % игр на тренировках с ботом у всех пользоватлей равноспределенно среди пользователей всех типов, а также среднее время на размышления, их можно удалить

In [184]:

```
```

245 / 4000

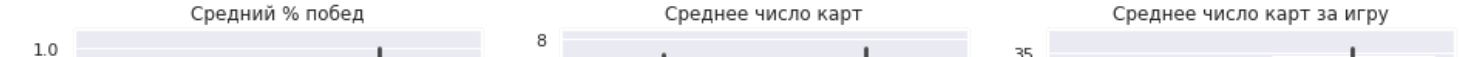


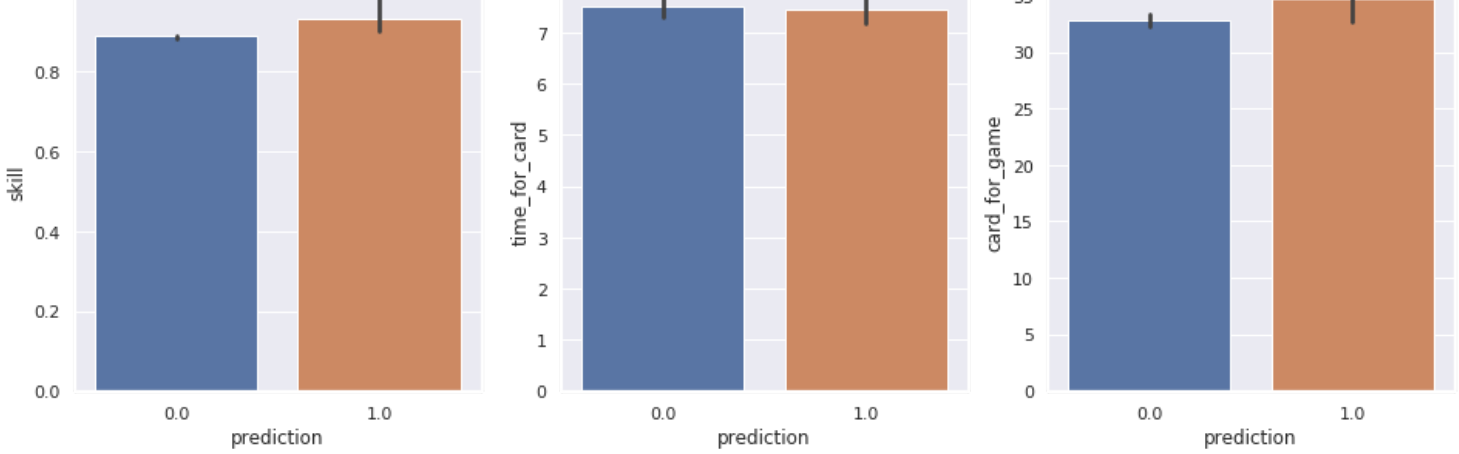
Роль среднего этапа соревнования на всеобщем турнире

Действительно, есть корреляция, причем чем выше игрок всреднем забирается по турнирной таблице, тем более его стратегия игры "нексомонавтская", причем для финального раунда различие почти в 2 раза. Не космонавты такде почти в 3 раза чаще побеждают во всем турнире

In [185]:

```
```

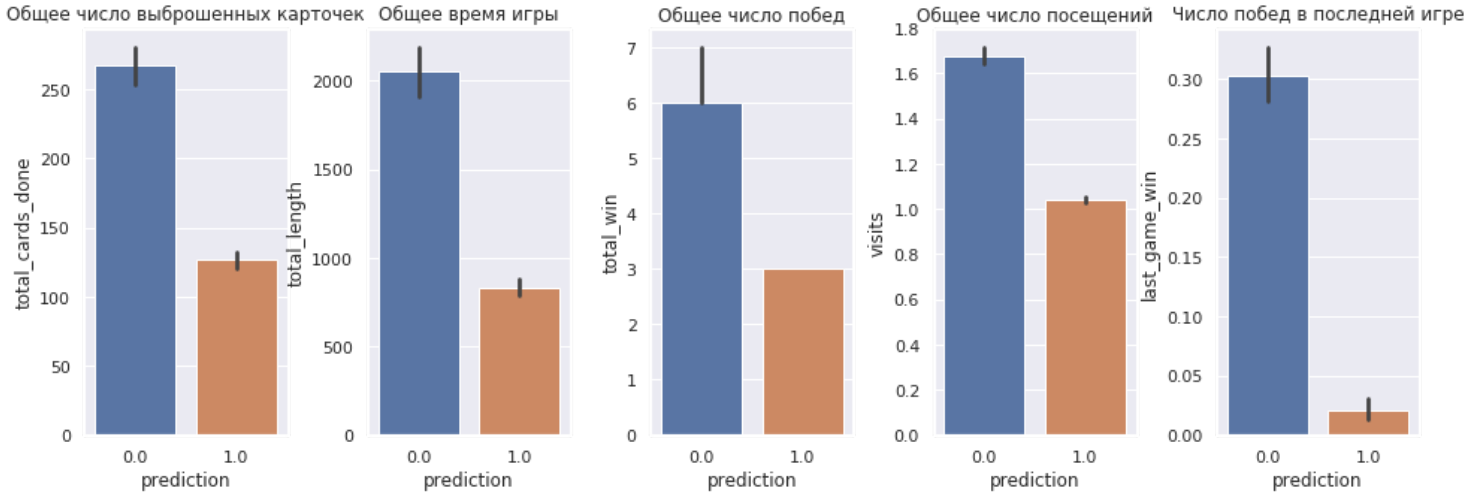




Роль относительных значений по параметрам игрока

Средние значения ни на что не влияют :(Вероятно слабым игрокам игра дает слабых соперников и все тратят +- одинаковое чило карт

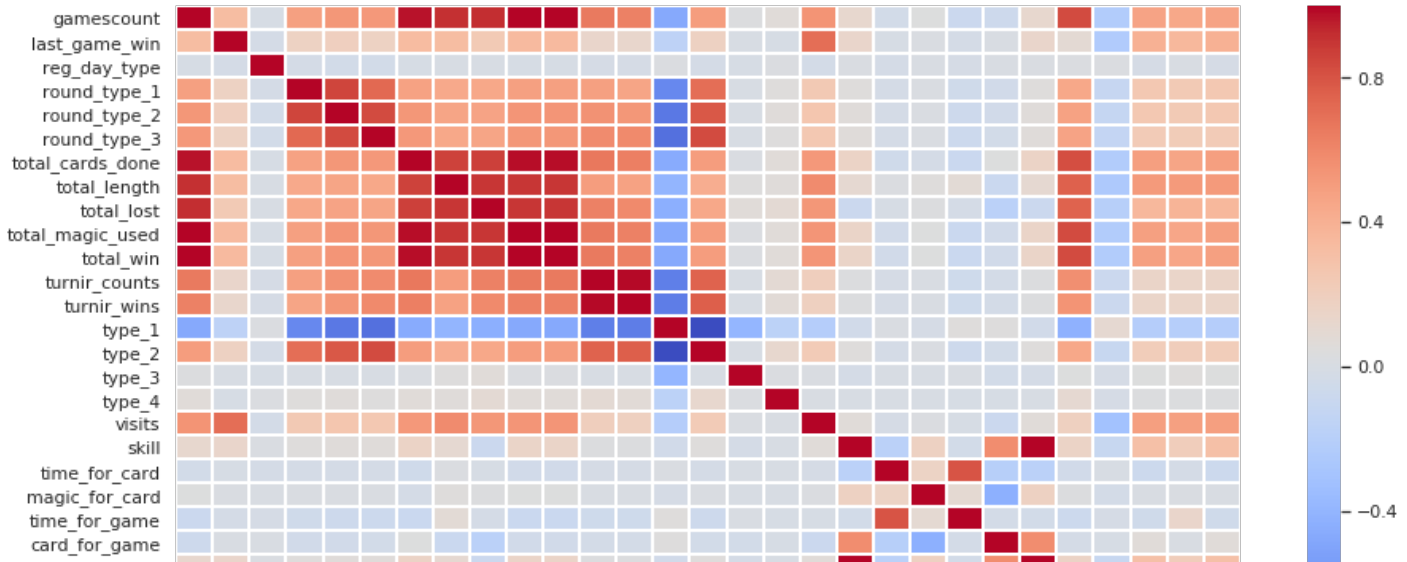
In [186]:

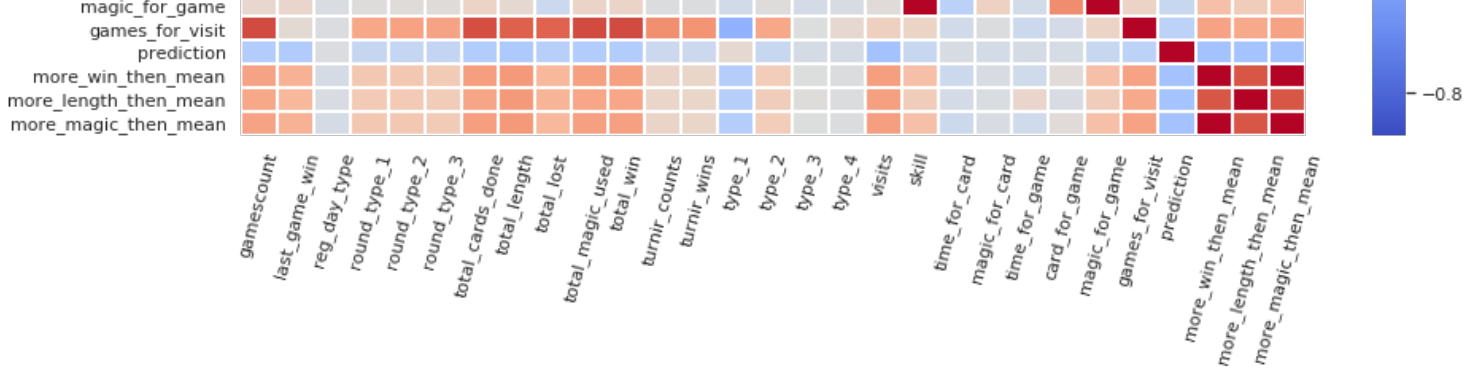


Посморим на абсолютные значений

Воу, медианные по абсолютным значениям отличаются на порядок в зависимости от космонавтности, добавим признаки больше / меньше среднемедианных значений по столбцам. Также заметим, что если игрок за сутки заходил в игру более 1 раза (разность времени между матчами была больше часа) то он почти 100% не космонавт. Также интересно, что у космонавтов очень мало побед в последнем матче, практически все проигрывали и выходили, вероятно растроившись и забыв.

In [188]:





Итоговый heatmap выводы

Сильная кореляция с новыми данными. Космонавтность связана с общим числом игр и соотв времени, проведенно в игре, количеством раундов и так далее. Низкая роль средних показателей вероятно связана с тем, что новичкам игра дает более простые стартовые сеты карт, а также более слабых противников на турнире. Опыт также коррелирует с навыками игры в чемпионатах и желанием игрока принимать в них участие. Можно сформулировать определение: **космонавт** - новичек в игре, человек который зашел потыкать, сыграл пару раундов и забил, не выработов свою стратегию игры и не достигнув значимого результата. Если эти выводы верны, выделенных ниже признаков будет доствточно, чтобы с высокой точностью предсказывть статус игрока.

In [191]:

```
Итоговые данные
```

Out[191]:

id	gamescount	last_game_win	round_type_1	round_type_2	round_type_3	total_cards_done	total_length	total_lost	total_magic_used	total_wir
218490	39	0	0.0	0.0	0.0	1008	21572	8	31	3
218492	3	0	0.0	0.0	0.0	108	949	0	3	3
218493	7	0	0.0	0.0	0.0	290	3298	3	4	4
218499	3	0	0.0	0.0	0.0	73	347	0	3	3
218507	4	0	0.0	0.0	0.0	156	724	1	3	3

5 rows × 23 columns

P.S. Несколько раз перечитал условия задачи - использовать признак космонавтности при проведении EAT нигде не запрещенно

In [272]:

```
```

Out[272]:

id	gamescount	last_game_win	round_type_1	round_type_2	round_type_3	total_cards_done	total_length	total_lost	total_magic_used	total_wir
218490	39	0	0.0	0.0	0.0	1008	21572	8	31	3
218493	7	0	0.0	0.0	0.0	290	3298	3	4	4
218499	3	0	0.0	0.0	0.0	73	347	0	3	3
218507	4	0	0.0	0.0	0.0	156	724	1	3	3
218508	12	1	0.0	0.0	0.0	438	3447	1	11	1

5 rows × 24 columns

In [261]:

```
collections.Counter(y_train)
```

Out[261]:

Counter({0.0: 2841, 1.0: 1159})

Классы несбалансированны, поэтому перед обучением модели сделаем oversampling

Нормализуем данные, хотя catboost вроде в любом случае сделает это за нас

Разделим данные на обучающую и тестовые выборки

Learning

Для классификации будем использовать градиентный бустинг по решающим деревьям из библиотеки catboost от яндекса, при анализе данных было выявлено несколько зависимостей итогвого значения от абсолютных значений, вособенности от к-ва заходов в игру, уровнях в турнирах и числа сыгранных матчей, которые должны отлично выявляться алгоритмами такого типа. После нескольких эксперементов с `sklearn.ensemble.RandomForestClassifier` и `XGBRFRegressor` лучшие результаты показала именно моделька от яндекса

Выберем наиболее удачные параметры для обучения, тк выделять данные на eval_set для автоматического выбора наилучшей модели, с учетом и без того малой выборки, было бы слишком расточительно

In [246]:

```
from sklearn.model_selection import GridSearchCV
from catboost import CatBoostClassifier


model = CatBoostClassifier(
    thread_count=4
)

parametrs = { 'iterations': range(400, 900, 50),
              'depth': range(3,7, 1),}

grid = GridSearchCV(model, parametrs, cv=5)
grid.fit(X_train, y_train, verbose=0)
grid.best_params_
```

Out[246]:

{'depth': 6, 'iterations': 850}

Обучим модельку с этими параметрами для ценки качества ее работы

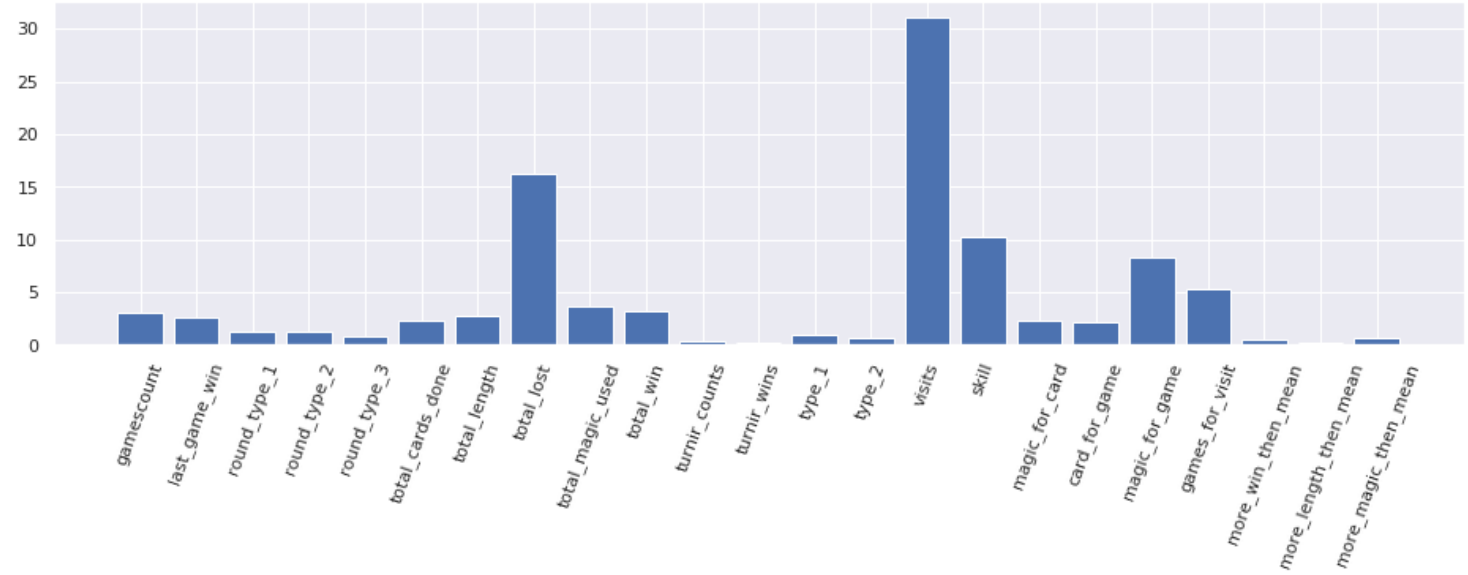
In [266]:

```
import matplotlib.pyplot as plt
```

```
import matplotlib.pyplot as plt
sns.set(style="darkgrid")

fig = plt.figure()
fig.set_size_inches(16, 4)

plt.bar(np.arange(len(model.feature_importances_)),model.feature_importances_)
plt.xticks(np.arange(len(model.feature_importances_)), model.feature_names_,rotation=70);
```



Как и ожидалось, наиболее важными параметрами оказались абосолютные значения игроков в игре

Посмотрим на `f1-score` нашей модели

In [267]:

```
from sklearn import metrics

y_pred = [model.predict(x) for x in X_validation.values]
print(metrics.classification_report(y_validation, y_pred,
                                    digits=3))
```

```
              precision    recall  f1-score   support

         0.0      0.922     0.626     0.746       433
         1.0      0.710     0.945     0.811       420

    accuracy                          0.783       853
   macro avg      0.816     0.786     0.778       853
weighted avg      0.818     0.783     0.778       853
```

Точность работы модельки: ~78%

Ну вот и все, спасибо за интересный кейс)