#### ИУ5-65Б Погосян С.Л. РК1

Bap №16

Номер задачи - 2, номер набора данных - 8

Рубежный контроль №1 Задание Для заданного набора данных проведите обработку пропусков в данных для одного категориального и одного количественного признака. Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали? Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему? Набор данных: <a href="https://www.kaggle.com/mathan/fifa-2018-match-statistics">https://www.kaggle.com/mathan/fifa-2018-match-statistics</a>

Дополнительное требование: Для студентов групп ИУ5-65Б — для набора данных построить «парные диаграммы»

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

## Загружаем данные:

```
In [2]:
      data = pd.read csv('FIFA 2018.csv', sep=",")
In [3]: # размер набора данных
       data.shape
Out[3]: (128, 27)
In [4]: # типы колонок
       data.dtypes
Out[4]: Date
                              object
       Team
                             object
       Opponent
                             object
       Goal Scored
                              int64
       Ball Possession %
                              int64
       Attempts
                              int64
       On-Target
                              int64
       Off-Target
                              int64
       Blocked
                              int64
       Corners
                              int64
       Offsides
                              int64
       Free Kicks
                              int64
                              int64
       Pass Accuracy %
                              int64
                              int64
       Distance Covered (Kms) int64
       Fouls Committed
                              int64
       Yellow Card
                              int64
       Yellow & Red
                              int64
                              int64
                           object
float64
       Man of the Match
       1st Goal
       Round
                             object
                             object
       Goals in PSO
                              int64
```

Own goals float64
Own goal Time float64

dtype: object

In [5]: # определим пропуски в столбцах data.isnull().sum()

Out[5]: Date 0 Team 0 Opponent 0 Goal Scored 0 Ball Possession % 0 0 Attempts 0 On-Target 0 Off-Target 0 Blocked 0 Corners Offsides Free Kicks Saves Pass Accuracy % Distance Covered (Kms) 0
Fouls Committed 0
Yellow Card 0 0 0 Yellow & Red Red Man of the Match 1st Goal 0 34 0 Round PSO Goals in PSO 0 116 116 Own goals Own goal Time dtype: int64

In [6]: # Первые 10 строк датасета data.head(10)

## Out[6]:

	Date	Team	Opponent	Goal Scored	Ball Possession %	Attempts	On- Target	Off- Target	Blocked	Corne
0	14- 06- 2018	Russia	Saudi Arabia	5	40	13	7	3	3	
1	14- 06- 2018	Saudi Arabia	Russia	0	60	6	0	3	3	
2	15- 06- 2018	Egypt	Uruguay	0	43	8	3	3	2	
3	15- 06- 2018	Uruguay	Egypt	1	57	14	4	6	4	
4	15- 06- 2018	Morocco	Iran	0	64	13	3	6	4	
5	15- 06- 2018	Iran	Morocco	1	36	8	2	5	1	
6	15- 06- 2018	Portugal	Spain	3	39	8	3	2	3	

7	15- 06- 2018	Spain	Portugal	3	61	12	5	5	2	
8	16- 06- 2018	France	Australia	2	51	12	5	4	3	
9	16- 06- 2018	Australia	France	1	49	4	1	2	1	

10 rows × 27 columns

```
In [7]: total_count = data.shape[0]
  print('Bcero ctpok: {}'.format(total_count))
```

Всего строк: 128

14-

Russia

# Обработка пропуков в числовых данных

```
In [8]: # Выберем числовые колонки с пропущенными значениями
# Цикл по колонкам датасета
num_cols = []
for col in data.columns:
# Количество пустых значений
temp_null_count = data[data[col].isnull()].shape[0]
dt = str(data[col].dtype)
if temp_null_count>0 and (dt=='float64' or dt=='int64'):
num_cols.append(col)
temp_perc = round((temp_null_count / total_count) * 100.0, 2)
print('Колонка {}. Тип данных {}. Количество пустых значений {},
```

Колонка 1st Goal. Тип данных float64. Количество пустых значений 34, 26.5 6%.

Колонка Own goals. Тип данных float64. Количество пустых значений 116, 9 0.62%.

Колонка Own goal Time. Тип данных float64. Количество пустых значений 11 6, 90.62%.

В колонках Own goals и Own goals содержится информация о наличии забитых голов в свои ворота и времени, когда это было сделано. Будем считать, что отсутствие информации говорит о том, что не было мячей забитых в свои ворота. Поэтому заполним пропуски в этих колонках нулями.

```
In [9]:
           # Заполнение всех пропущенных значений в "Own goals" нулями
           data new 3 = data[['Own goals']].fillna(0)
          data[['Own goals']] = data_new_3
In [10]:
           # Заполнение всех пропущенных значений в "Own goal Time" нулями
In [11]:
           data new 4 = data[['Own goal Time']].fillna(0)
          data[['Own goal Time']]=data new 4
In [12]:
In [13]:
           data.head()
                                                 Ball
Out[13]:
                                     Goal
                                                                On-
                                                                       Off-
                    Team Opponent
                                          Possession Attempts
                                                                           Blocked Corne
             Date
                                   Scored
                                                              Target Target
                                                  %
```

5

40

13

7

3

Saudi

3

	06- 2018		Arabia						
1	14- 06- 2018	Saudi Arabia	Russia	0	60	6	0	3	3
2	15- 06- 2018	Egypt	Uruguay	0	43	8	3	3	2
3	15- 06- 2018	Uruguay	Egypt	1	57	14	4	6	4
4	15- 06- 2018	Morocco	Iran	0	64	13	3	6	4

5 rows × 27 columns

В колонке 1st goal содержится информация о времени от начала игры, в которое был забит первый гол. Поэтому заполним пропуски нулями в тех строках, где значение "Goal Scored" ранво нулю.

```
In [14]: data.loc[(data['Goal Scored'] == 0), '1st Goal'] = 0
```

Заполним оставшиеся пропуски в столбце "1st Goal" средним значением по столбцу.

```
data[['1st Goal']].describe()
In [15]:
                   1st Goal
Out[15]:
          count 127.000000
           mean
                  29.204724
                  27.289552
            std
            min
                   0.000000
            25%
                   0.000000
            50%
                  24.000000
            75%
                  51.000000
                  90.000000
            max
           res = np.where(np.isnan(data['1st Goal']), np.ma.array(data['1st Goal'],
In [16]:
```

```
mask = np.isnan(data['1st Goal'])).mean(axis = 0), data['
In [17]: data['1st Goal']=res
In [18]: data.head()
```

Out[18]:		Date	Team	Opponent	Goal Scored	Ball Possession %	Attempts	On- Target	Off- Target	Blocked	Corne
	0	14- 06- 2018	Russia	Saudi Arabia	5	40		7		3	
	1	14-	Saudi	Russia	0	60	6	0	3	3	

	06- 2018	Arabia							
2	15- 06- 2018	Egypt	Uruguay	0	43	8	3	3	2
3	15- 06- 2018	Uruguay	Egypt	1	57	14	4	6	4
4	15- 06- 2018	Morocco	Iran	0	64	13	3	6	4

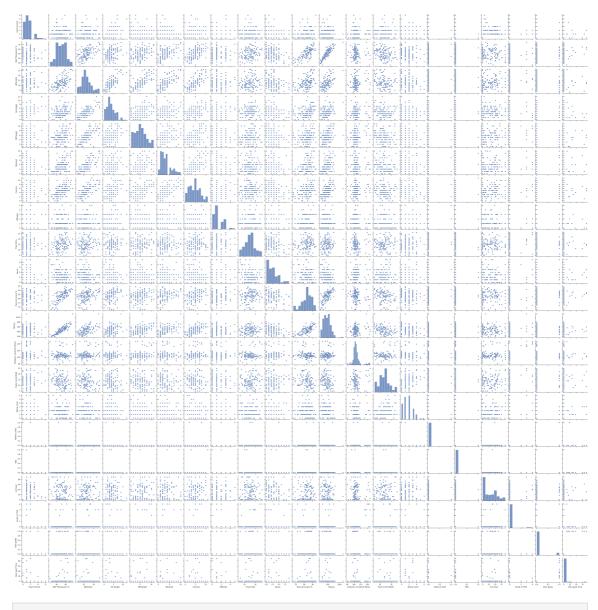
### 5 rows × 27 columns

```
In [19]: # проверим пропуски в столбцах
         data.isnull().sum()
Out[19]: Date
                                  0
         Team
         Opponent
         Goal Scored
         Ball Possession %
         Attempts
         On-Target
        Off-Target
         Blocked
         Corners
         Offsides
         Free Kicks
         Saves
         Pass Accuracy %
         Passes
        Distance Covered (Kms)
         Fouls Committed
         Yellow Card
         Yellow & Red
         Red
        Man of the Match
        1st Goal
         Round
         PSO
         Goals in PSO
                                 0
         Own goals
                                 0
         Own goal Time
         dtype: int64
```

# Парные диаграммы

```
In [20]: sns.pairplot(data)
```

Out[20]: <seaborn.axisgrid.PairGrid at 0x7f658236d0a0>



In []: