ИУ5-62Б Перова А.Е. Рубежный контроль No1 (вариант 16)

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?

Подключение библиотек для анализа данных

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV, RandomizedSearchCV
from sklearn.neighbors import KNeighborsRegressor
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from matplotlib import pyplot as plt
import seaborn as sns
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from warnings import simplefilter
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator

from sklearn.impute import MissingIndicator
##warnings.simplefilter('ignore')
```

Загрузка датасета из файла `FIFA 2018 Statistics.csv`

```
data = pd.read_csv('FIFA 2018 Statistics.csv')
```

Проверка данных

```
data.head()
```

	Date	Team	Opponent	Goal Scored	Ball Possession %	Attempts	On-Target	Off-Target	Blocked	Corners	...	Yellow Card	Yellow & Red	Red	Man of the Match	1st Goal	Round	PSO	Goals in PSO	Ov goa
0	14-06-2018	Russia	Saudi Arabia	5	40	13	7	3	3	6	...	0	0	0	Yes	12.0	Group Stage	No	0	Na
1	14-06-2018	Saudi Arabia	Russia	0	60	6	0	3	3	2	...	0	0	0	No	NaN	Group Stage	No	0	Na
2	15-06-2018	Egypt	Uruguay	0	43	8	3	3	2	0	...	2	0	0	No	NaN	Group Stage	No	0	Na
3	15-06-2018	Uruguay	Egypt	1	57	14	4	6	4	5	...	0	0	0	Yes	89.0	Group Stage	No	0	Na
4	15-06-2018	Morocco	Iran	0	64	13	3	6	4	5	...	1	0	0	No	NaN	Group Stage	No	0	1

5 rows × 27 columns

Видим, что данные загружены корректно. Разбиения по строкам и столбцам произведены верно. Проблем с кодировкой не возникло. Узнаем размер датасета:

```
print(f'Количество записей: {data.shape[0]}\nКоличество параметров: {data.shape[1]}')
```

```
Количество записей: 128
Количество параметров: 27
```

Очистка данных

Посмотрим краткую информацию обо всех параматрах датасета:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 128 entries, 0 to 127
Data columns (total 27 columns):
Date                        128 non-null object
Team                        128 non-null object
Opponent                    128 non-null object
Goal Scored                 128 non-null int64
Ball Possession %           128 non-null int64
Attempts                    128 non-null int64
On-Target                   128 non-null int64
Off-Target                  128 non-null int64
Blocked                     128 non-null int64
Corners                     128 non-null int64
Offsides                    128 non-null int64
Free Kicks                  128 non-null int64
Saves                       128 non-null int64
Pass Accuracy %             128 non-null int64
Passes                      128 non-null int64
Distance Covered (Kms)      128 non-null int64
Fouls Committed             128 non-null int64
Yellow Card                 128 non-null int64
Yellow & Red                128 non-null int64
Red                         128 non-null int64
Man of the Match            128 non-null object
1st Goal                    94 non-null float64
Round                       128 non-null object
PSO                         128 non-null object
Goals in PSO                128 non-null int64
Own goals                   12 non-null float64
Own goal Time               12 non-null float64
dtypes: float64(3), int64(18), object(6)
memory usage: 27.1+ KB
```

Видим, что в датасете присутствуют нулевые значения. выведем список параметров датасета и для каждого из них найдём количество `null` значений.

```
for column in data.columns:
    print(f'{column}: {data[column].isnull().sum()} null values')
```

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

Заметим, что столбцы Own goals и Own goal Time имеют большинство (116 из 128) строк, поэтому удалим эти столбцы

```
data = data.drop(['Own goals'], axis =1)
```

```
data = data.drop(['Own goal Time'], axis =1)
```

```
data.head()
```

```
data.head()
```

	Date	Team	Opponent	Goal Scored	Ball Possession %	Attempts	On-Target	Off-Target	Blocked	Corners	...	Distance Covered (Kms)	Fouls Committed	Yellow Card	Yellow & Red	Red	Man of the Match	1st Goal	Ro
0	14-06-2018	Russia	Saudi Arabia	5	40	13	7	3	3	6	...	118	22	0	0	0	Yes	12.0	G S
1	14-06-2018	Saudi Arabia	Russia	0	60	6	0	3	3	2	...	105	10	0	0	0	No	NaN	G S
2	15-06-2018	Egypt	Uruguay	0	43	8	3	3	2	0	...	112	12	2	0	0	No	NaN	G S
3	15-06-2018	Uruguay	Egypt	1	57	14	4	6	4	5	...	111	6	0	0	0	Yes	89.0	G S
4	15-06-2018	Morocco	Iran	0	64	13	3	6	4	5	...	101	22	1	0	0	No	NaN	G S

5 rows × 25 columns

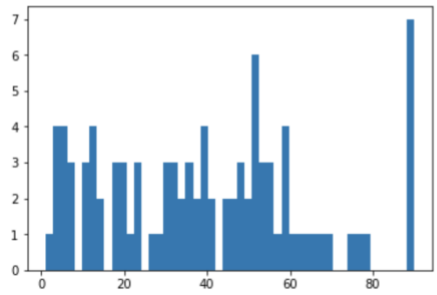
```
total_count = data.shape[0]
num_cols = []
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='float64' or dt=='int64'):
        num_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.format(col, dt, temp_null_count, temp_p
erc))
```

```
Колонка 1st Goal. Тип данных float64. Количество пустых значений 34, 26.56%.
```

```
data_num = data[num_cols]
data_num
```

	1st Goal
0	12.0
1	NaN
2	NaN
3	89.0
4	NaN
5	90.0
6	4.0
7	24.0
8	58.0
9	62.0
10	19.0
11	23.0
12	NaN
13	59.0
14	32.0
15	NaN
16	NaN
17	56.0
18	NaN
19	35.0
20	20.0
21	50.0

```
data[np.isfinite(data['1st Goal'])]
#plt.hist(data[np.isfinite(data['1st Goal'])].values,50)
plt.hist(data['1st Goal'].dropna(),50)
plt.show()
```




```
data_num_Goal = data_num[['1st Goal']]
data_num_Goal.head()
```

	1st Goal
0	12.0
1	NaN
2	NaN
3	89.0
4	NaN

```
# Фильтр для проверки заполнения пустых значений
indicator = MissingIndicator()
mask_missing_values_only = indicator.fit_transform(data_num_Goal)
mask_missing_values_only
```

```
def test_num_impute(param):
    imp_num = SimpleImputer(strategy=param)
    data_num_imp = imp_num.fit_transform(data_num_Goal)
    return data_num_imp[mask_missing_values_only]
```

```
def test_num_impute(strategy_param):
    imp_num = SimpleImputer(strategy=strategy_param)
    data_num_imp = imp_num.fit_transform(data_num_MasVnrArea)
    return data_num_imp[mask_missing_values_only]
```

```
test_num_impute('median')
```

```
array([39., 39., 39., 39., 39., 39., 39., 39., 39., 39., 39., 39.,
       39., 39., 39., 39., 39., 39., 39., 39., 39., 39., 39., 39.,
       39., 39., 39., 39., 39., 39., 39., 39.])
```

В дальнейшем для обучения модели можно взять категориальные признаки: Team,Opponent, Man of the Match числовые признаки: Goal scored, Ball Posession, Attempts, On-Target, Off-Targed,Fouls Committed, так как они будут иметь наибольшее влияние на предсказание результатов матчей