РК1 вариант 16

Пшенин, РТ5-61Б

Файл по заданию: https://www.kaggle.com/mathan/fifa-2018-match-statistics

Задание: для заданного набора данных провести обработку пропусков в данных для одного категориального и одного количественного признака. Какие способы обработки пропусков в данных для категориальных и количественных признаков использовались? Какие признаки можно использовать для дальнейшего построения моделей машинного обучения и почему? Дополнительное задание: построить Jointplot.

In [4]:

```
import numpy as np
import pandas as pd
fifa = pd.read_csv('C:\\FIFA 2018 Statistics.csv')
```

In [6]:

```
fifa.head(3)
```

Out[6]:

	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
0	14-06-2018	Russia	Saudi Arabia	5	40	13	7	3	3	6	...	0	0	0	Yes	12.0	Group Stage	No
1	14-06-2018	Saudi Arabia	Russia	0	60	6	0	3	3	2	...	0	0	0	No	NaN	Group Stage	No
2	15-06-2018	Egypt	Uruguay	0	43	8	3	3	2	0	...	2	0	0	No	NaN	Group Stage	No

3 rows × 27 columns

In [8]:

```
fifa.shape
```

Out[8]:

```
(128, 27)
```

In [9]:

```
fifa.dtypes
```

Out[9]:

```
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
Saves                       int64
Pass Accuracy %             int64
Passes                      int64
Distance Covered (Kms)      int64
Fouls Committed             int64
Yellow Card                 int64
Yellow & Red                int64
Red                         int64
Man of the Match           object
1st Goal                  float64
Round                      object
PSO                        object
Goals in PSO                int64
Own goals                 float64
Own goal Time             float64
dtype: object
```

In [10]:

```
fifa.isnull().sum()
```

Out[10]:

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

In [11]:

```
34 / 128, 116 / 128
```

Out[11]:

```
(0.265625, 0.90625)
```

Выводы о колонках с пропусками и их влиянии на модель: Из 27 колонок пропуски есть только в трёх. Допустимым будем считать число пропусков в колонке 1st Goal (27%). Колонки Own goal и Own goal Time не будем включать в модель, так как количество пропущенных данных в них превышает 90%.

Заполнение пропусков в данных

Для начала отбросим колонки Own goal и Own goal Time.

In [13]:

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

Теперь заполним пропуски в колонке 1st Goal. Это числовая колонка, будем использовать SimpleImputer со стратегией "среднее арифметическое".

In [20]:

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
```

In [21]:

```
fifa['1st Goal'].describe()
```

Out[21]:

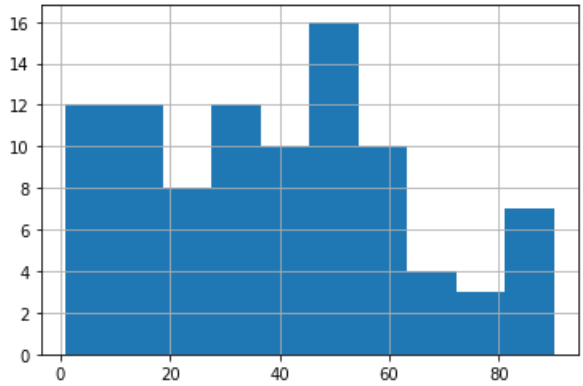
```
count    94.000000
mean     39.457447
std      24.496506
min       1.000000
25%      18.250000
50%      39.000000
75%      54.750000
max      90.000000
Name: 1st Goal, dtype: float64
```

In [24]:

```
fifa['1st Goal'].hist()
```

Out[24]:

```
<AxesSubplot:>
```



Думаю, нет никаких особых причин менять стратегию на моду или медиану из-за формы гистограммы.

In [25]:

```
fifa['1st Goal'] = imputer.fit_transform(fifa[['1st Goal']])
```

In [28]:

```
fifa['1st Goal'].isnull().sum()
```

Out[28]:

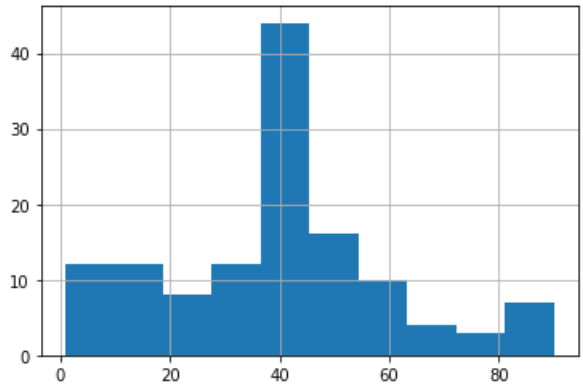
```
0
```

In [29]:

```
fifa['1st Goal'].hist()
```

Out[29]:

```
<AxesSubplot:>
```



Итак, пропусков в данных больше нет, но и распределение изменилось, как можно заметить.

JointPlot

In [32]:

```
import seaborn as sns
sns.jointplot(data=fifa,x="Ball Possession %",y="Goal Scored")
```

Out[32]:

```
<seaborn.axisgrid.JointGrid at 0x1b786371310>
```