Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

ДИСЦИПЛИНА Технологии машинного обучения

Отчёт
по рубежному контролю № 1

"Технологии разведочного анализа и обработки данных"

Вариант 9

Выполнил:

Студент группы ИУ5-63 _____ **Королев С.В.**

(Подпись, дата) (Фамилия И.О.)

Проверил:

_____ **Гапанюк Ю.Е.**

(Подпись, дата) (Фамилия И.О.)

Москва, 2020

Рубежный контроль №1

"Технологии разведочного анализа и обработки данных"

Вариант 9

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?

Использованный набор данных

https://www.kaggle.com/karangadiya/fifa19 (https://www.kaggle.com/karangadiya/fifa19)

In [126]:
```
import numpy as np
import pandas as pd
pd.set_option('display.max.columns', 100)
# to draw pictures in jupyter notebook
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
# we don't like warnings
# you can comment the following 2 lines if you'd like to
import warnings
warnings.filterwarnings('ignore')
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

In [127]:
```
# Таблица данных
data = pd.read_csv('data.csv')
data.head()
```

Out[127]:

	Unnamed: 0	ID	Name	Age	Photo	Nationality	Flag	Overa
0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentina	https://cdn.sofifa.org/flags/52.png	9
1	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portugal	https://cdn.sofifa.org/flags/38.png	9
2	2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Brazil	https://cdn.sofifa.org/flags/54.png	9
3	3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spain	https://cdn.sofifa.org/flags/45.png	9
4	4	192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgium	https://cdn.sofifa.org/flags/7.png	9

Выведем основные параметры этого набора данных

In [128]:
```
# Размер набора данных
data.shape
```

Out[128]: (18207, 89)

In [129]:
```
# Типы данных в колонках
data.dtypes
```

Out[129]:

Unnamed: 0	int64
ID	int64
Name	object
Age	int64
Photo	object
Nationality	object
Flag	object
Overall	int64
Potential	int64
Club	object
Club Logo	object
Value	object
Wage	object
Special	int64
Preferred Foot	object
International Reputation	float64
Weak Foot	float64
Skill Moves	float64
Work Rate	object
Body Type	object
Real Face	object
Position	object
Jersey Number	float64
Joined	object
Loaned From	object
Contract Valid Until	object
Height	object
Weight	object
LS	object
ST	object
RS	object
LW	object
LF	object
CF	object
RF	object
RW	object
LAM	object
CAM	object
RAM	object
LM	object
LCM	object
CM	object
RCM	object
RM	object
LWB	object
LDM	object
CDM	object
RDM	object
RWB	object
LB	object
LCB	object
CB	object
RCB	object
RB	object
Crossing	float64
Finishing	float64

Количество пропусков в данных

In [130]:
```
data.isnull().sum()
```

Out[130]:

Unnamed: 0	0
ID	0
Name	0
Age	0
Photo	0
Nationality	0
Flag	0
Overall	0
Potential	0
Club	241
Club Logo	0
Value	0
Wage	0
Special	0
Preferred Foot	48
International Reputation	48
Weak Foot	48
Skill Moves	48
Work Rate	48
Body Type	48
Real Face	48
Position	60
Jersey Number	60
Joined	1553
Loaned From	16943
Contract Valid Until	289
Height	48
Weight	48
LS	2085
ST	2085
RS	2085
LW	2085
LF	2085
CF	2085
RF	2085
RW	2085
LAM	2085
CAM	2085
RAM	2085
LM	2085
LCM	2085
CM	2085
RCM	2085
RM	2085
LWB	2085
LDM	2085
CDM	2085
RDM	2085
RWB	2085
LB	2085
LCB	2085
CB	2085
RCB	2085
RB	2085
Crossing	48
Finishing	48

Обработка пропусков в данных

1. Количественный признак

Займемся обработкой пропусков в колонке "Crossing" - Вероятность успеха паса напереред

In [131]:
```
data['Crossing'].unique()
```

Out[131]:
```
array([84., 79., 17., 93., 81., 86., 77., 66., 13., 62., 88., 55., 6
8.,
       82., 75., 15., 14., 70., 58., 78., 52., 90., 64., 87., 60., 1
2.,
       69., 46., 30., 80., 11., 57., 83., 85., 20., 73., 53., 72., 3
6.,
       44., 45., 40., 27., 76., 63., 65., 48., 61., 47., 89., 19., 4
9.,
        9., 71., 74., 24., 18., 38., 92., 56., 67., 35., 25., 50., 2
9.,
       10., 42., 54., 59., 91., 51., 43., 33., 34., 16., 39., 28.,
8.,
       21., 23., 41., 32., 37., 31., 22.,  7., 26.,  6.,  5., nan])
```

In [132]:
```
# Преобразование типа колонок с пропущенными числовыми значениями в чи
словой
data['Crossing'] = data['Crossing'].astype(float)
```

Выведем статистику по пропущенным значениям в колонках

In [133]:
```
# Количество пустых значений
col = 'Crossing'
temp_null_count = data[data[col].isnull()].shape[0]
total_count = data.shape[0]
dt = str(data[col].dtype)
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_perc))
```

```
Колонка Crossing. Тип данных float64. Количество пустых значений 48,
0.26%.
```

Заполнение пропусков в столбце

In [134]:
```
# Фильтр по пустым значениям поля normalized-losses
data[data[col].isnull()]
```

Out[134]:

	Unnamed: 0	ID	Name	Age	Photo	Nationality	Flag
13236	13236	177971	J. McNulty	33	https://cdn.sofifa.org/players/4/19/177971.png	Scotland	https://cdn.sofifa.org/flags/42.png
13237	13237	195380	J. Barrera	29	https://cdn.sofifa.org/players/4/19/195380.png	Nicaragua	https://cdn.sofifa.org/flags/86.png
13238	13238	139317	J. Stead	35	https://cdn.sofifa.org/players/4/19/139317.png	England	https://cdn.sofifa.org/flags/14.png
13239	13239	240437	A. Semprini	20	https://cdn.sofifa.org/players/4/19/240437.png	Italy	https://cdn.sofifa.org/flags/27.png
13240	13240	209462	R. Bingham	24	https://cdn.sofifa.org/players/4/19/209462.png	England	https://cdn.sofifa.org/flags/14.png
13241	13241	219702	K. Dankowski	21	https://cdn.sofifa.org/players/4/19/219702.png	Poland	https://cdn.sofifa.org/flags/37.png
13242	13242	225590	I. Colman	23	https://cdn.sofifa.org/players/4/19/225590.png	Argentina	https://cdn.sofifa.org/flags/52.png
13243	13243	233782	M. Feeney	19	https://cdn.sofifa.org/players/4/19/233782.png	England	https://cdn.sofifa.org/flags/14.png
13244	13244	239158	R. Minor	30	https://cdn.sofifa.org/players/4/19/239158.png	Denmark	https://cdn.sofifa.org/flags/13.png
13245	13245	242998	Klauss	21	https://cdn.sofifa.org/players/4/19/242998.png	Brazil	https://cdn.sofifa.org/flags/54.png
13246	13246	244022	I. Sissoko	22	https://cdn.sofifa.org/players/4/19/244022.png	France	https://cdn.sofifa.org/flags/18.png
13247	13247	189238	F. Hart	28	https://cdn.sofifa.org/players/4/19/189238.png	Austria	https://cdn.sofifa.org/flags/4.png
13248	13248	211511	L. McCullough	24	https://cdn.sofifa.org/players/4/19/211511.png	Northern Ireland	https://cdn.sofifa.org/flags/35.png
13249	13249	224055	Li Yunqiu	27	https://cdn.sofifa.org/players/4/19/224055.png	China PR	https://cdn.sofifa.org/flags/155.png
13250	13250	244535	F. Garcia	29	https://cdn.sofifa.org/players/4/19/244535.png	Paraguay	https://cdn.sofifa.org/flags/58.png
13251	13251	134968	R. Haemhouts	34	https://cdn.sofifa.org/players/4/19/134968.png	Belgium	https://cdn.sofifa.org/flags/7.png

In [135]:
```
# Запоминаем индексы строк с пустыми значениями
flt_index = data[data[col].isnull()].index
flt_index
```

Out[135]:
```
Int64Index([13236, 13237, 13238, 13239, 13240, 13241, 13242, 13243, 1
3244,
            13245, 13246, 13247, 13248, 13249, 13250, 13251, 13252, 1
3253,
            13254, 13255, 13256, 13257, 13258, 13259, 13260, 13261, 1
3262,
            13263, 13264, 13265, 13266, 13267, 13268, 13269, 13270, 1
3271,
            13272, 13273, 13274, 13275, 13276, 13277, 13278, 13279, 1
3280,
            13281, 13282, 13283],
           dtype='int64')
```

In [136]:
```
# фильтр по колонке
data[data.index.isin(flt_index)][col]
```

Out[136]:
```
13236   NaN
13237   NaN
13238   NaN
13239   NaN
13240   NaN
13241   NaN
13242   NaN
13243   NaN
13244   NaN
13245   NaN
13246   NaN
13247   NaN
13248   NaN
13249   NaN
13250   NaN
13251   NaN
13252   NaN
13253   NaN
13254   NaN
13255   NaN
13256   NaN
13257   NaN
13258   NaN
13259   NaN
13260   NaN
13261   NaN
13262   NaN
13263   NaN
13264   NaN
13265   NaN
13266   NaN
13267   NaN
13268   NaN
13269   NaN
13270   NaN
13271   NaN
13272   NaN
13273   NaN
13274   NaN
13275   NaN
13276   NaN
13277   NaN
13278   NaN
13279   NaN
13280   NaN
13281   NaN
13282   NaN
13283   NaN
Name: Crossing, dtype: float64
```

Будем использовать встроенные средства импьютации библиотеки scikit-learn - https://scikit-learn.org/stable/modules/impute.html#impute (https://scikit-learn.org/stable/modules/impute.html#impute)

In [137]:
```
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
```

In [138]:
```
data_cross = data[col]
data_cross
```

Out[138]:

0	84.0
1	84.0
2	79.0
3	17.0
4	93.0
5	81.0
6	86.0
7	77.0
8	66.0
9	13.0
10	62.0
11	88.0
12	55.0
13	84.0
14	68.0
15	82.0
16	75.0
17	82.0
18	15.0
19	14.0
20	62.0
21	70.0
22	15.0
23	70.0
24	58.0
25	77.0
26	78.0
27	52.0
28	90.0
29	86.0
30	75.0
31	88.0
32	79.0
33	77.0
34	64.0
35	90.0
36	87.0
37	13.0
38	68.0
39	60.0
40	12.0
41	13.0
42	69.0
43	46.0
44	30.0
45	80.0
46	11.0
47	78.0
48	55.0
49	87.0
50	77.0
51	68.0
52	78.0
53	84.0
54	57.0
55	83.0

С помощью класса SimpleImputer проведем импьютацию с различными показателями центра распределения ("среднее", "медиана", "самое частое")

In [139]:
```
def test_num_impute_col(dataset, column, strategy_param):
    temp_data = dataset[[column]]

    indicator = MissingIndicator()
    mask_missing_values_only = indicator.fit_transform(temp_data)

    imp_num = SimpleImputer(strategy = strategy_param)
    data_num_imp = imp_num.fit_transform(temp_data)

    filled_data = data_num_imp[mask_missing_values_only]

    return column, strategy_param, filled_data.size, filled_data[0], f
illed_data[filled_data.size-1]
```

In [140]:
```
# центр распределения - "среднее"
test_num_impute_col(data, col, "mean")
```

Out[140]: ('Crossing', 'mean', 48, 49.734181397654055, 49.734181397654055)

In [141]:
```
# центр распределения - "медиана"
test_num_impute_col(data, col, "median")
```

Out[141]: ('Crossing', 'median', 48, 54.0, 54.0)

In [142]:
```
# центр распределения - "самое частое"
test_num_impute_col(data, col, "most_frequent")
```

Out[142]: ('Crossing', 'most_frequent', 48, 62.0, 62.0)

Таким образом можно избавиться от пропусков во всех колонках, но это происходит весьма неосознанно

Осознанное заполнение пропусков

Из описания набора данных известно, что значат все признаки. Учитывая их взаимосвязь между собой, можно попробовать приблизительно посчитать возможные значения пропущенных данных.

Раз этот показатель вычисляется среди игроков определнной классификации, то мы можем попробовать заменить пропуски данных в этой колонке средним значением показателя "Potential"

In [143]:
```
for row in data.itertuples():
     if np.isnan(row.Crossing):
            data_potential = data.loc[data['Potential']== row.Potentia
l]
            data.at[row.Index, col] = data[col].mean()
data.isnull().sum()
```

Out[143]:

Unnamed: 0	0
ID	0
Name	0
Age	0
Photo	0
Nationality	0
Flag	0
Overall	0
Potential	0
Club	241
Club Logo	0
Value	0
Wage	0
Special	0
Preferred Foot	48
International Reputation	48
Weak Foot	48
Skill Moves	48
Work Rate	48
Body Type	48
Real Face	48
Position	60
Jersey Number	60
Joined	1553
Loaned From	16943
Contract Valid Until	289
Height	48
Weight	48
LS	2085
ST	2085
RS	2085
LW	2085
LF	2085
CF	2085
RF	2085
RW	2085
LAM	2085
CAM	2085
RAM	2085
LM	2085
LCM	2085
CM	2085
RCM	2085
RM	2085
LWB	2085
LDM	2085
CDM	2085
RDM	2085
RWB	2085
LB	2085
LCB	2085
CB	2085
RCB	2085
RB	2085
Crossing	0
Finishing	48

Таким образом, мы убрали все пропуски в "Crossing"

2. Категориальный признак

Заполнение пропусков в столбце "club"

In [144]:
```
# Преобразование типа колонок с пропущенными числовыми значениями в ст
роковый
data['Club'] = data['Club'].astype(str)
```

In [145]:
```
# Выберем даннее только из этой колонки
col_club = 'Club'
club_data = data[[col_club]]
club_data.head()
```

Out[145]:

	Club
0	FC Barcelona
1	Juventus
2	Paris Saint-Germain
3	Manchester United
4	Manchester City

In [146]:
```
# Все возможнве клубы (все уникальные значения колонки)
data[col_club].unique()
```

```
Out[146]: array(['FC Barcelona', 'Juventus', 'Paris Saint-Germain',
                 'Manchester United', 'Manchester City', 'Chelsea', 'Real Madri
          d',
                 'Atlético Madrid', 'FC Bayern München', 'Tottenham Hotspur',
                 'Liverpool', 'Napoli', 'Arsenal', 'Milan', 'Inter', 'Lazio',
                 'Borussia Dortmund', 'Vissel Kobe', 'Olympique Lyonnais', 'Rom
          a',
                 'Valencia CF', 'Guangzhou Evergrande Taobao FC', 'FC Porto',
                 'FC Schalke 04', 'Beşiktaş JK', 'LA Galaxy', 'Sporting CP',
                 'Real Betis', 'Olympique de Marseille', 'RC Celta',
                 'Bayer 04 Leverkusen', 'Real Sociedad', 'Villarreal CF',
                 'Sevilla FC', 'SL Benfica', 'AS Saint-Étienne', 'AS Monaco',
                 'Leicester City', 'Atalanta', 'Grêmio', 'Atlético Mineiro',
                 'RB Leipzig', 'Ajax', 'Dalian YiFang FC', 'Everton',
                 'West Ham United', '1. FC Köln', 'TSG 1899 Hoffenheim',
                 'Shanghai SIPG FC', 'OGC Nice', 'Al Nassr',
                 'Wolverhampton Wanderers', 'Borussia Mönchengladbach',
                 'Hertha BSC', 'SV Werder Bremen', 'Cruzeiro',
                 'Athletic Club de Bilbao', 'Torino', 'Medipol Başakşehir FK',
                 'Beijing Sinobo Guoan FC', 'Crystal Palace', 'PFC CSKA Moscow
          ',
                 'VfL Wolfsburg', 'Shakhtar Donetsk', 'Toronto FC',
                 'Lokomotiv Moscow', 'Sassuolo', 'New York City FC', 'Fluminens
          e',
                 'PSV', 'Levante UD', 'Fulham', 'Watford', 'Atlanta United',
                 'Montpellier HSC', 'Galatasaray SK', 'Fenerbahçe SK', 'SD Eiba
          r',
                 'Los Angeles FC', 'Sampdoria', 'Al Hilal', 'VfB Stuttgart',
                 'SC Braga', 'River Plate', 'Deportivo Alavés', 'nan',
                 'Eintracht Frankfurt', 'Girona FC', 'Guangzhou R&F; FC', 'Burn
          ley',
                 'Stoke City', 'Southampton', 'Tianjin Quanjian FC', 'Getafe CF
          ',
                 'Beijing Renhe FC', 'Montreal Impact', 'Chievo Verona', 'Genoa
          ',
                 'Portland Timbers', 'Tigres U.A.N.L.', 'RCD Espanyol',
                 'Hebei China Fortune FC', 'Cagliari', 'Chicago Fire', 'DC Unit
          ed',
                 'Sagan Tosu', 'Dynamo Kyiv', 'Santos', 'Internacional',
                 'América FC (Minas Gerais)', 'Independiente', 'Boca Juniors',
                 'Cruz Azul', '1. FSV Mainz 05', 'Bournemouth', 'Spartak Moscow
          ',
                 'Racing Club', 'FC Augsburg', 'Fiorentina', 'FC Nantes',
                 'Feyenoord', 'Club Brugge KV', 'Brighton & Hove Albion', 'Al A
          hli',
                 'Jiangsu Suning FC', 'SC Freiburg', 'PAOK', 'Stade Rennais FC
          ',
                 'Trabzonspor', 'SPAL', 'Portimonense SC', 'Olympiacos CFP',
                 'Club Atlético Huracán', 'Kasimpaşa SK', 'Newcastle United',
                 'Frosinone', 'Querétaro', 'KRC Genk', 'Hannover 96',
                 'Stade Malherbe Caen', 'Godoy Cruz', 'Toulouse Football Club',
                 'RSC Anderlecht', 'Huddersfield Town', 'CD Tondela',
                 'Seattle Sounders FC', 'Hamburger SV', 'FC Red Bull Salzburg',
                 'Rio Ave FC', 'FC Girondins de Bordeaux', 'Melbourne Victory',
                 'Parma', 'FC Basel 1893', 'Al Wehda', 'BSC Young Boys', 'KAA G
          ent',
```

In [147]:
```
# Размер колонки
data[col_club].unique().shape
```

Out[147]:
```
(652,)
```

In [148]:
```
for row in data.itertuples():
     if pd.isnull(row.Club):
            Nationality_data = data.loc[data['Nationality'] == row.Nat
ionality]
            if (pd.isnull(Nationality_data['Club'].mode(dropna = Fals
e)[0])):
                data.at[row.Index, 'Club'] = data['Club'].mode()[0] #
Если и это не помогло, то просто находим самое популярное по всей табл
ице:
            else:
                data.at[row.Index, 'Club'] = Nationality_data['Club'].
mode()[0]
data.isnull().sum()
```

Out[148]:

Unnamed: 0	0
ID	0
Name	0
Age	0
Photo	0
Nationality	0
Flag	0
Overall	0
Potential	0
Club	0
Club Logo	0
Value	0
Wage	0
Special	0
Preferred Foot	48
International Reputation	48
Weak Foot	48
Skill Moves	48
Work Rate	48
Body Type	48
Real Face	48
Position	60
Jersey Number	60
Joined	1553
Loaned From	16943
Contract Valid Until	289
Height	48
Weight	48
LS	2085
ST	2085
RS	2085
LW	2085
LF	2085
CF	2085
RF	2085
RW	2085
LAM	2085
CAM	2085
RAM	2085
LM	2085
LCM	2085
CM	2085
RCM	2085
RM	2085
LWB	2085
LDM	2085
CDM	2085
RDM	2085
RWB	2085
LB	2085
LCB	2085
CB	2085
RCB	2085
RB	2085
Crossing	0
Finishing	48

Таким образом, мы избавились от пропусков в категориальных данных.

In [ ]: