

Светашева Юлия ИУ5-64Б

17 вариант РК-1

Номер задачи - 3, номер набора данных – 1.

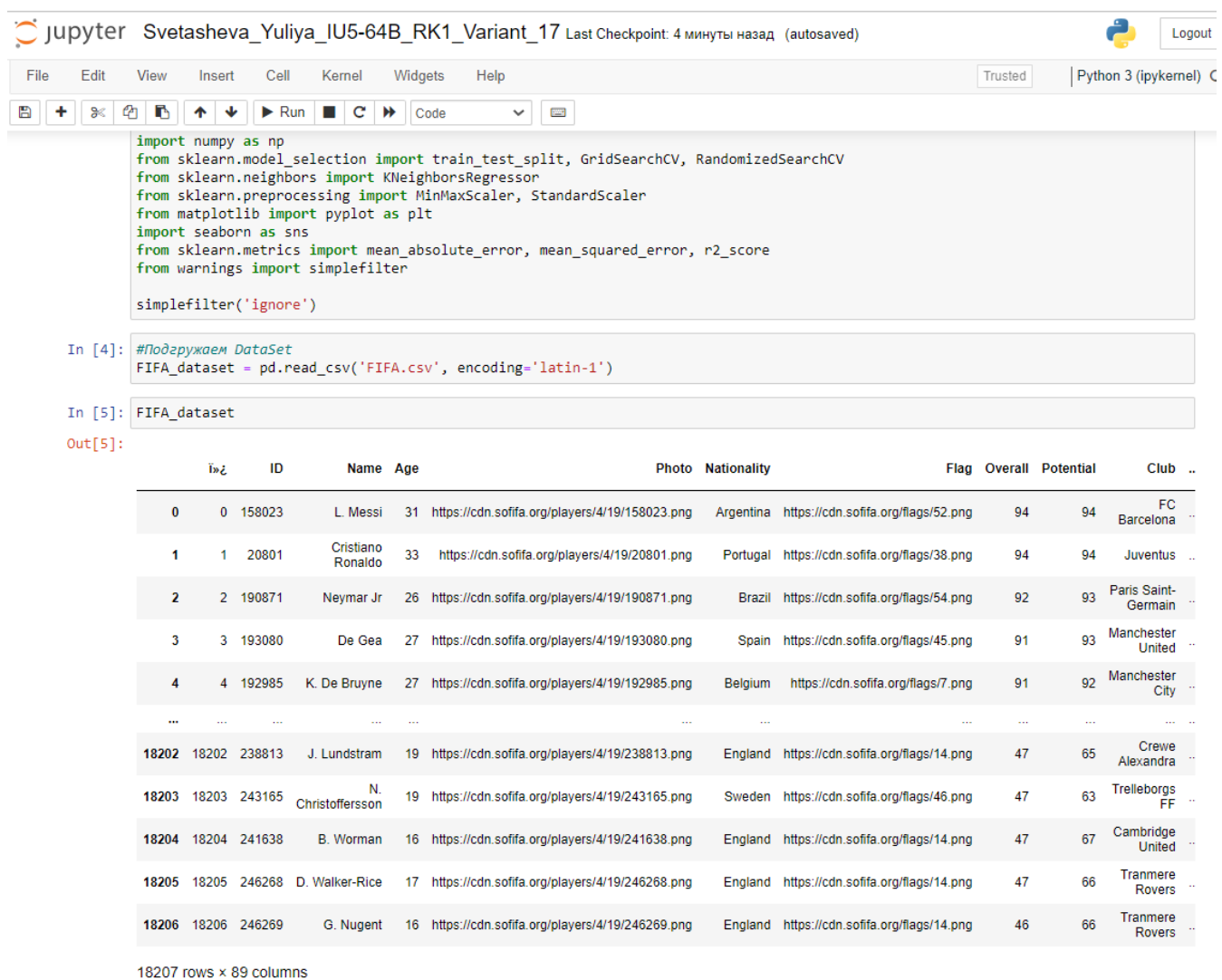
Для студентов группы ИУ5-64Б, ИУ5Ц-84Б - для произвольной колонки данных построить график "Скрипичная диаграмма (violin plot)".

Задача №3.

Для заданного набора данных произведите масштабирование данных (для одного признака) и преобразование категориальных признаков в количественные двумя способами (label encoding, one hot encoding) для одного признака. Какие методы Вы использовали для решения задачи и почему?

Используемый набор данных: [FIFA 19 complete player dataset | Kaggle](#)

Подгружаем необходимые библиотеки и датасет:



```
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

simplefilter('ignore')

In [4]: #Подгружаем DataSet
FIFA_dataset = pd.read_csv('FIFA.csv', encoding='latin-1')

In [5]: FIFA_dataset

Out[5]:
```

ID	Name	Age	Photo	Nationality	Flag	Overall	Potential	Club
0	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentina	https://cdn.sofifa.org/flags/52.png	94	94	FC Barcelona
1	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portugal	https://cdn.sofifa.org/flags/38.png	94	94	Juventus
2	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Brazil	https://cdn.sofifa.org/flags/54.png	92	93	Paris Saint-Germain
3	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spain	https://cdn.sofifa.org/flags/45.png	91	93	Manchester United
4	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgium	https://cdn.sofifa.org/flags/7.png	91	92	Manchester City
...
18202	J. Lundstram	19	https://cdn.sofifa.org/players/4/19/238813.png	England	https://cdn.sofifa.org/flags/14.png	47	65	Crewe Alexandra
18203	N. Christoffersson	19	https://cdn.sofifa.org/players/4/19/243165.png	Sweden	https://cdn.sofifa.org/flags/46.png	47	63	Trelleborgs FF
18204	B. Worman	16	https://cdn.sofifa.org/players/4/19/241638.png	England	https://cdn.sofifa.org/flags/14.png	47	67	Cambridge United
18205	D. Walker-Rice	17	https://cdn.sofifa.org/players/4/19/246268.png	England	https://cdn.sofifa.org/flags/14.png	47	66	Tranmere Rovers
18206	G. Nugent	16	https://cdn.sofifa.org/players/4/19/246269.png	England	https://cdn.sofifa.org/flags/14.png	46	66	Tranmere Rovers

18207 rows x 89 columns

Выводим информацию о столбцах датасета:



Code



In [6]:



```
<class 'pandas.core.frame.DataFrame'>
```





RangeIndex: 18207 entries, 0 to 18206

Data columns (total 89 columns):

#	Column	Non-Null Count	Dtype
0	i»¿	18207 non-null	int64
1	ID	18207 non-null	int64
2	Name	18207 non-null	object
3	Age	18207 non-null	int64
4	Photo	18207 non-null	object
5	Nationality	18207 non-null	object
6	Flag	18207 non-null	object
7	Overall	18207 non-null	int64
8	Potential	18207 non-null	int64
9	Club	17966 non-null	object
10	Club Logo	18207 non-null	object
11	Value	18207 non-null	object
12	Wage	18207 non-null	object
13	Special	18207 non-null	int64
14	Preferred Foot	18159 non-null	object
15	International Reputation	18159 non-null	float64
16	Weak Foot	18159 non-null	float64
17	Skill Moves	18159 non-null	float64
18	Work Rate	18159 non-null	object
19	Body Type	18159 non-null	object
20	Real Face	18159 non-null	object
21	Position	18147 non-null	object
22	Jersey Number	18147 non-null	float64
23	Joined	16654 non-null	object
24	Loaned From	1264 non-null	object
25	Contract Valid Until	17918 non-null	object
26	Height	18159 non-null	object
27	Weight	18159 non-null	object
28	LS	16122 non-null	object
29	ST	16122 non-null	object
30	RS	16122 non-null	object
31	LW	16122 non-null	object
32	LF	16122 non-null	object
33	CF	16122 non-null	object
34	RF	16122 non-null	object
35	RW	16122 non-null	object
36	LAM	16122 non-null	object
37	CAM	16122 non-null	object
38	RAM	16122 non-null	object
39	LM	16122 non-null	object

File Edit View Insert Cell Kernel Widgets Help




 Run
 


 Code

```

39 LM 16122 non-null object
40 LCM 16122 non-null object
41 CM 16122 non-null object
42 RCM 16122 non-null object
43 RM 16122 non-null object
44 LWB 16122 non-null object
45 LDM 16122 non-null object
46 CDM 16122 non-null object
47 RDM 16122 non-null object
48 RWB 16122 non-null object
49 LB 16122 non-null object
50 LCB 16122 non-null object
51 CB 16122 non-null object
52 RCB 16122 non-null object
53 RB 16122 non-null object
54 Crossing 18159 non-null float64
55 Finishing 18159 non-null float64
56 HeadingAccuracy 18159 non-null float64
57 ShortPassing 18159 non-null float64
58 Volleys 18159 non-null float64
59 Dribbling 18159 non-null float64
60 Curve 18159 non-null float64
61 FKAccuracy 18159 non-null float64
62 LongPassing 18159 non-null float64
63 BallControl 18159 non-null float64
64 Acceleration 18159 non-null float64
65 SprintSpeed 18159 non-null float64
66 Agility 18159 non-null float64
67 Reactions 18159 non-null float64
68 Balance 18159 non-null float64
69 ShotPower 18159 non-null float64
70 Jumping 18159 non-null float64
71 Stamina 18159 non-null float64
72 Strength 18159 non-null float64
73 LongShots 18159 non-null float64
74 Aggression 18159 non-null float64
75 Interceptions 18159 non-null float64
76 Positioning 18159 non-null float64
77 Vision 18159 non-null float64
78 Penalties 18159 non-null float64
79 Composure 18159 non-null float64
80 Marking 18159 non-null float64
81 StandingTackle 18159 non-null float64
82 SlidingTackle 18159 non-null float64
83 GKDividing 18159 non-null float64
84 GKHandling 18159 non-null float64
85 GKKicking 18159 non-null float64
86 GKPositioning 18159 non-null float64
87 GKReflexes 18159 non-null float64
88 Release Clause 16643 non-null object
dtypes: float64(38), int64(6), object(45)
    
```

```
In [5]: #категоральный признаков в датасете слишком много, они будут мешать делать масштабирование данных.  
#Оставим только те столбцы, с которыми потом будем работать  
FIFA_dataset_new = FIFA_dataset[['ID', 'Age', 'Overall', 'Potential', 'Name', 'Photo', 'Nationality', 'Flag']].copy()  
FIFA_dataset_new
```

18207 rows x 8 columns

	ID	Age	Overall	Potential	Name_cat	Photo_cat	Nationality_cat	Flag_cat
0	158023	31	94	94	9632	566	6	122
1	20801	33	94	94	3153	6031	123	107
2	190871	26	92	93	12508	3131	20	124
3	193080	27	91	93	4136	3467	139	114
4	192985	27	91	92	8617	3452	13	137
...
18202	238813	19	47	65	7580	14347	46	40
18203	243165	19	47	63	12101	16252	144	115
18204	241638	16	47	67	2133	15506	46	40
18205	246268	17	47	66	3997	17998	46	40
18206	246269	16	46	66	5807	17999	46	40


Разделяем выборки

```
In [8]: #разделение выборки
from sklearn.model_selection import train_test_split
y = FIFA_dataset_new['Age']
X = FIFA_dataset_new.drop('Age', axis=1)
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=3)
x_train
```


Out[8]:

	ID	Overall	Potential	Name_cat	Photo_cat	Nationality_cat	Flag_cat
6291	201138	69	72	17019	4614	55	68
12013	181491	64	64	12081	1814	8	78
9753	241115	66	78	13508	15287	158	131
12705	239523	63	71	11429	14627	59	85
5004	203588	70	70	13466	5134	37	151
...
6400	195020	69	70	3378	3776	46	40
15288	237818	60	71	12027	14023	6	122
11513	240511	64	77	6537	15041	13	137
1688	214076	75	75	12460	7488	158	131
5994	192611	69	69	15319	3389	97	27

12744 rows × 7 columns

 jupyter Svetasheva_Yuliya_IU5-64B_RK1_Variant_17 Last Checkpoint

File Edit View Insert Cell Kernel Widgets Help



In [9]: y_train

Out[9]:

6291	26
12013	30
9753	21
12705	23
5004	27
...	..
6400	27
15288	20
11513	19
1688	34
5994	29

Name: Age, Length: 12744, dtype: int64

Масштабирование данных

jupyter Svetasheva_Yuliya_IU5-64B_RK1_Variant_17 Last Checkpoint: 5 минут назад (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Code

```
15288 20
11513 19
1688 34
5994 29
Name: Age, Length: 12744, dtype: int64
```

In [10]: *#Масштабирование данных*

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler().fit(x_train)
x_train = pd.DataFrame(scaler.transform(x_train), columns = x_train.columns)
x_test = pd.DataFrame(scaler.transform(x_test), columns = x_train.columns)
x_train.describe()
```

Out[10]:

	ID	Overall	Potential	Name_cat	Photo_cat	Nationality_cat	Flag_cat
count	12744.000000	12744.000000	12744.000000	12744.000000	12744.000000	12744.000000	12744.000000
mean	0.868855	0.421430	0.494871	0.498019	0.500269	0.467034	0.560302
std	0.121885	0.143599	0.129891	0.287196	0.288367	0.294190	0.227823
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.812567	0.333333	0.404255	0.250785	0.250275	0.214724	0.386503
50%	0.899541	0.416667	0.489362	0.494561	0.501565	0.361963	0.631902
75%	0.959086	0.520833	0.574468	0.744547	0.750055	0.754601	0.748466
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Обучение KNN с производным k

jupyter Svetasheva_Yuliya_IU5-64B_RK1_Variant_17 Last Checkpoint: 6 минут назад (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipy)

```
75% 0.959086 0.520833 0.574468 0.744547 0.750055 0.754601 0.748466
max 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
```

In [11]: *#Обучение KNN с производным k*

```
simplefilter('ignore')


def print_metrics(y_test, y_pred):
    print(f"R^2: {r2_score(y_test, y_pred)}")
    print(f"MSE: {mean_squared_error(y_test, y_pred)}")
    print(f"MAE: {mean_absolute_error(y_test, y_pred)}")

def print_cv_result(cv_model, x_test, y_test):
    print(f'Оптимизация метрики {cv_model.scoring}: {cv_model.best_score_}')
    print(f'Лучший параметр: {cv_model.best_params_}')
    print('Метрики на тестовом наборе')
    print_metrics(y_test, cv_model.predict(x_test))
    print()
base_k = 7
base_knn = KNeighborsRegressor(n_neighbors=base_k)
base_knn.fit(x_train, y_train)
y_pred_base = base_knn.predict(x_test)
print(f'Test metrics for KNN with k={base_k}\n')
print_metrics(y_test, y_pred_base)

Test metrics for KNN with k=7

R^2: 0.8166653972347018
MSE: 3.9743767908041856
MAE: 1.462121806438116
```


Кросс-валидация

 jupyter Svetasheva_Yuliya_IU5-64B_RK1_Variant_17 Last Checkpoint: 8 минут назад (autosaved)

File Edit View Insert Cell Kernel Widgets Help

       Run    Code 

```
In [12]: #Кросс валидация
metrics = ['r2', 'neg_mean_squared_error', 'neg_mean_absolute_error']
cv_values = [5, 10]

for cv in cv_values:
    print(f'Результаты кросс-валидации при cv={cv}\n')
    for metric in metrics:
        params = {'n_neighbors': range(1, 30)}
        knn_cv = GridSearchCV(KNeighborsRegressor(), params, cv=cv, scoring=metric, n_jobs=-1)
        knn_cv.fit(x_train, y_train)
        print_cv_result(knn_cv, x_test, y_test)
```

Результаты кросс-валидации при cv=5

Оптимизация метрики r2: 0.8095840296479139

Лучший параметр: {'n_neighbors': 6}

Метрики на тестовом наборе

R^2: 0.8144504474035854

MSE: 4.02239306852157

MAE: 1.4703764720239185

Оптимизация метрики neg_mean_squared_error: -4.160861419338133

Лучший параметр: {'n_neighbors': 6}

Метрики на тестовом наборе

R^2: 0.8144504474035854

MSE: 4.02239306852157

MAE: 1.4703764720239185

Оптимизация метрики neg_mean_absolute_error: -1.5084448113675264

Лучший параметр: {'n_neighbors': 6}

Метрики на тестовом наборе

R^2: 0.8144504474035854

MSE: 4.02239306852157

MAE: 1.4703764720239185

Результаты кросс-валидации при cv=10

Оптимизация метрики r2: 0.8125512978309442

Лучший параметр: {'n_neighbors': 8}

Метрики на тестовом наборе

R^2: 0.8162869124407597

MSE: 3.9825816858868754

MAE: 1.460781621819513

Оптимизация метрики neg_mean_squared_error: -4.092533715409241

Лучший параметр: {'n_neighbors': 8}

Метрики на тестовом наборе

R^2: 0.8162869124407597

MSE: 3.9825816858868754

MAE: 1.460781621819513

Оптимизация метрики neg_mean_absolute_error: -1.4938646689445008

Лучший параметр: {'n_neighbors': 8}

Метрики на тестовом наборе

R^2: 0.8162869124407597

MSE: 3.9825816858868754

MAE: 1.460781621819513

Скрипичная диаграмма по столбцу «Age»

```
In [18]: sns.violinplot(x=FIFA_dataset_new['Age'])
```

```
Out[18]: <AxesSubplot:xlabel='Age'>
```

