

Импорт библиотек

```
In [95]: import numpy as np #библиотека для работы с многомерными массивами данных и математическими операциями над ними
import pandas as pd #библиотека для анализа и обработки данных
from typing import Dict, Tuple
from scipy import stats
import matplotlib.pyplot as plt #простое рисование графиков
import seaborn as sns #удобные дефолтные настройки графиков из matplotlib
from gmdhpy import gmdh
from gmdhpy.gmdh import MultilayerGMDH
from IPython.display import Image
from sklearn.preprocessing import MinMaxScaler, LabelEncoder, OneHotEncoder
from sklearn.model_selection import cross_val_score, cross_validate, train_test_split, GridSearchCV, RandomizedSearchCV, KFold, RepeatedKFold, LeaveOneOut, LeavePOut, ShuffleSplit, StratifiedKFold, learning_curve, validation_curve
from sklearn.neighbors import KNeighborsRegressor, KNeighborsClassifier
from sklearn.metrics import accuracy_score, mean_absolute_error, mean_squared_error, mean_squared_log_error, median_absolute_error, precision_score, recall_score, f1_score, classification_report, confusion_matrix, roc_curve, roc_auc_score
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor, export_graphviz
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor, ExtraTreesClassifier, ExtraTreesRegressor, GradientBoostingClassifier, GradientBoostingRegressor, BaggingClassifier, AdaBoostRegressor
from sklearn.svm import SVC, NuSVC, LinearSVC, OneClassSVM, SVR, NuSVR, LinearSVR
from sklearn.linear_model import LinearRegression
from sklearn.impute import SimpleImputer
import lightgbm # сожрет все сырьим и построит регрессионную модель, которая покажет важные фичи
# чтобы дальше делать лабу только на них

%matplotlib inline
#для сохранения в ноутбуке вывода моих графиков
```

```
In [100]: def print_stat(test_target, real_target):
    print(mean_absolute_error(test_target, real_target))
    print(mean_squared_error(test_target, real_target))
    print(median_absolute_error(test_target, real_target))
```

Загрузка данных

```
In [43]: data = pd.read_csv('./data/fifa19.csv')
data = data.drop(['ID', 'Photo', 'Flag', 'Club Logo'], axis=1)
data
```

Out[43]:

	Unnamed: 0	Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Special	...	Composure	Marking	StandingTa
0	0	L. Messi	31	Argentina	94	94	FC Barcelona	€110.5M	€565K	2202	...	96.0	33.0	
1	1	Cristiano Ronaldo	33	Portugal	94	94	Juventus	€77M	€405K	2228	...	95.0	28.0	
2	2	Neymar Jr	26	Brazil	92	93	Paris Saint-Germain	€118.5M	€290K	2143	...	94.0	27.0	
3	3	De Gea	27	Spain	91	93	Manchester United	€72M	€260K	1471	...	68.0	15.0	
4	4	K. De Bruyne	27	Belgium	91	92	Manchester City	€102M	€355K	2281	...	88.0	68.0	
5	5	E. Hazard	27	Belgium	91	91	Chelsea	€93M	€340K	2142	...	91.0	34.0	
6	6	L. Modrić	32	Croatia	91	91	Real Madrid	€67M	€420K	2280	...	84.0	60.0	
7	7	L. Suárez	31	Uruguay	91	91	FC Barcelona	€80M	€455K	2346	...	85.0	62.0	
8	8	Sergio Ramos	32	Spain	91	91	Real Madrid	€51M	€380K	2201	...	82.0	87.0	
9	9	J. Oblak	25	Slovenia	90	93	Atlético Madrid	€68M	€94K	1331	...	70.0	27.0	
10	10	R. Lewandowski	29	Poland	90	90	FC Bayern München	€77M	€205K	2152	...	86.0	34.0	
11	11	T. Kroos	28	Germany	90	90	Real Madrid	€76.5M	€355K	2190	...	85.0	72.0	
12	12	D. Godín	32	Uruguay	90	90	Atlético Madrid	€44M	€125K	1946	...	82.0	90.0	
13	13	David Silva	32	Spain	90	90	Manchester City	€60M	€285K	2115	...	93.0	59.0	
14	14	N. Kanté	27	France	89	90	Chelsea	€63M	€225K	2189	...	85.0	90.0	
15	15	P. Dybala	24	Argentina	89	94	Juventus	€89M	€205K	2092	...	84.0	23.0	
16	16	H. Kane	24	England	89	91	Tottenham Hotspur	€83.5M	€205K	2165	...	89.0	56.0	
17	17	A. Griezmann	27	France	89	90	Atlético Madrid	€78M	€145K	2246	...	87.0	59.0	

Unnamed: 0		Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Special	...	Composure	Marking	StandingTa
18	18	M. ter Stegen	26	Germany	89	92	FC Barcelona	€58M	€240K	1328	...	69.0	25.0	
19	19	T. Courtois	26	Belgium	89	90	Real Madrid	€53.5M	€240K	1311	...	66.0	20.0	
20	20	Sergio Busquets	29	Spain	89	89	FC Barcelona	€51.5M	€315K	2065	...	90.0	90.0	
21	21	E. Cavani	31	Uruguay	89	89	Paris Saint- Germain	€60M	€200K	2161	...	82.0	52.0	
22	22	M. Neuer	32	Germany	89	89	FC Bayern München	€38M	€130K	1473	...	70.0	17.0	
23	23	S. Agüero	30	Argentina	89	89	Manchester City	€64.5M	€300K	2107	...	90.0	30.0	
24	24	G. Chiellini	33	Italy	89	89	Juventus	€27M	€215K	1841	...	84.0	93.0	
25	25	K. Mbappé	19	France	88	95	Paris Saint- Germain	€81M	€100K	2118	...	86.0	34.0	
26	26	M. Salah	26	Egypt	88	89	Liverpool	€69.5M	€255K	2146	...	91.0	38.0	
27	27	Casemiro	26	Brazil	88	90	Real Madrid	€59.5M	€285K	2170	...	84.0	88.0	
28	28	J. Rodríguez	26	Colombia	88	89	FC Bayern München	€69.5M	€315K	2171	...	87.0	52.0	
29	29	L. Insigne	27	Italy	88	88	Napoli	€62M	€165K	2017	...	83.0	51.0	
...
18177	18177	R. Roache	18	Republic of Ireland	48	69	Blackpool	€70K	€1K	1178	...	49.0	18.0	
18178	18178	L. Wahlstedt	18	Sweden	48	65	Dalkurd FF	€50K	€1K	738	...	28.0	16.0	
18179	18179	J. Williams	17	England	48	64	Northampton Town	€50K	€1K	1166	...	37.0	42.0	
18180	18180	M. Hurst	22	Scotland	48	58	St. Johnstone FC	€40K	€1K	987	...	28.0	12.0	
18181	18181	C. Maher	17	Republic of Ireland	48	66	Bray Wanderers	€50K	€1K	1296	...	38.0	43.0	

Unnamed: 0		Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Special	...	Composure	Marking	StandingTa
18182	18182	Y. Gómez	18	Colombia	48	65	Atlético Nacional	€50K	€1K	1311	...	38.0	44.0	
18183	18183	K. Pilkington	44	England	48	48	Cambridge United	€0	€1K	774	...	56.0	15.0	
18184	18184	D. Horton	18	England	48	55	Lincoln City	€40K	€1K	1368	...	42.0	47.0	
18185	18185	E. Tweed	19	Republic of Ireland	48	59	Derry City	€50K	€1K	1315	...	43.0	39.0	
18186	18186	Zhang Yufeng	20	China PR	47	64	Beijing Renhe FC	€60K	€1K	1389	...	39.0	53.0	
18187	18187	C. Ehlich	19	Germany	47	59	SpVgg Unterhaching	€40K	€1K	1366	...	47.0	40.0	
18188	18188	L. Collins	17	Wales	47	62	Newport County	€60K	€1K	1297	...	46.0	33.0	
18189	18189	A. Kaltner	18	Germany	47	61	SpVgg Unterhaching	€60K	€1K	1290	...	37.0	28.0	
18190	18190	L. Watkins	18	England	47	67	Cambridge United	€60K	€1K	1285	...	46.0	35.0	
18191	18191	J. Norville-Williams	18	England	47	65	Cambridge United	€50K	€1K	1250	...	36.0	45.0	
18192	18192	S. Squire	18	England	47	64	Cambridge United	€50K	€1K	1325	...	38.0	41.0	
18193	18193	N. Fuentes	18	Chile	47	64	Unión Española	€50K	€1K	1191	...	32.0	41.0	
18194	18194	J. Millii	18	Italy	47	65	Lecce	€50K	€1K	731	...	23.0	6.0	
18195	18195	S. Griffin	18	Republic of Ireland	47	67	Waterford FC	€60K	€1K	1325	...	41.0	44.0	
18196	18196	K. Fujikawa	19	Japan	47	61	Júbilo Iwata	€60K	€1K	1333	...	35.0	41.0	
18197	18197	D. Holland	18	Republic of Ireland	47	61	Cork City	€60K	€1K	1362	...	52.0	41.0	
18198	18198	J. Livesey	18	England	47	70	Burton Albion	€60K	€1K	792	...	34.0	15.0	

	Unnamed: 0	Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Special	...	Composure	Marking	StandingTa
18199	18199	M. Baldisimo	18	Canada	47	69	Vancouver Whitecaps FC	€70K	€1K	1303	...	40.0	48.0	
18200	18200	J. Young	18	Scotland	47	62	Swindon Town	€60K	€1K	1203	...	50.0	15.0	
18201	18201	D. Walsh	18	Republic of Ireland	47	68	Waterford FC	€60K	€1K	1098	...	43.0	44.0	
18202	18202	J. Lundstram	19	England	47	65	Crewe Alexandra	€60K	€1K	1307	...	45.0	40.0	
18203	18203	N. Christoffersson	19	Sweden	47	63	Trelleborgs FF	€60K	€1K	1098	...	42.0	22.0	
18204	18204	B. Worman	16	England	47	67	Cambridge United	€60K	€1K	1189	...	41.0	32.0	
18205	18205	D. Walker-Rice	17	England	47	66	Tranmere Rovers	€60K	€1K	1228	...	46.0	20.0	
18206	18206	G. Nugent	16	England	46	66	Tranmere Rovers	€60K	€1K	1321	...	43.0	40.0	

18207 rows × 85 columns

Этот набор данных создан для прогнозирования Цены футболиста.

Содержание

Набор данных содержит несколько параметров, которые считаются важными во время оценки уровня счастья. Включенные параметры:

1. Имя
2. Возраст
3. Гражданство
4. Клуб
5. Характеристики (набор числовых значений в различных столбцах)
6. Позиция (один из нескольких столбцов)
7. Репутация
8. Название клуба, который его арендовал
- и др.

Целевой признак:

1. Цена

```
In [44]: data.shape
```

```
Out[44]: (18207, 85)
```

В нашем наборе данных 18207 строк и 89 столбцов. Посмотрим тип данных:

In [45]: `data.dtypes`

```
Out[45]: Unnamed: 0           int64
Name                  object
Age                   int64
Nationality          object
Overall               int64
Potential             int64
Club                  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
...
Dribbling             float64
Curve                 float64
FKAccuracy            float64
LongPassing            float64
BallControl             float64
Acceleration            float64
SprintSpeed             float64
Agility                 float64
Reactions               float64
Balance                 float64
```

```
ShotPower           float64
Jumping            float64
Stamina            float64
Strength           float64
LongShots          float64
Aggression         float64
Interceptions     float64
Positioning        float64
Vision              float64
Penalties           float64
Composure           float64
Marking             float64
StandingTackle      float64
SlidingTackle       float64
GKDiving            float64
GKHandling          float64
GKKicking            float64
GKPositioning       float64
GKReflexes          float64
Release Clause      object
Length: 85, dtype: object
```

Посмотрим, есть ли пропущенные значения в данных:

In [46]: `data.isnull().sum()`

Out[46]: Unnamed: 0

Name	0
Age	0
Nationality	0
Overall	0
Potential	0
Club	241
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
	...
Dribbling	48
Curve	48
FKAccuracy	48
LongPassing	48
BallControl	48
Acceleration	48
SprintSpeed	48
Agility	48
Reactions	48
Balance	48

ShotPower	48
Jumping	48
Stamina	48
Strength	48
LongShots	48
Aggression	48
Interceptions	48
Positioning	48
Vision	48
Penalties	48
Composure	48
Marking	48
StandingTackle	48
SlidingTackle	48
GKDiving	48
GKHandling	48
GKKicking	48
GKPositioning	48
GKReflexes	48
Release Clause	1564

Length: 85, dtype: int64

Если признак не числовой, сделаем его категориальным, а после - закодируем

```
In [47]: for column in data.select_dtypes(include = ['object']).columns.tolist():
    data[column] = data[column].astype('category')
data.head()
```

Out[47]:

		Unnamed: 0	Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Special	...	Composure	Marking	Standing	Tackle	Slidin
0	0	L. Messi	31	Argentina	94	94	FC Barcelona	€110.5M	€565K	2202	...	96.0	33.0		28.0		
1	1	Cristiano Ronaldo	33	Portugal	94	94	Juventus	€77M	€405K	2228	...	95.0	28.0		31.0		
2	2	Neymar Jr	26	Brazil	92	93	Paris Saint- Germain	€118.5M	€290K	2143	...	94.0	27.0		24.0		
3	3	De Gea	27	Spain	91	93	Manchester United	€72M	€260K	1471	...	68.0	15.0		21.0		
4	4	K. De Bruyne	27	Belgium	91	92	Manchester City	€102M	€355K	2281	...	88.0	68.0		58.0		

5 rows × 85 columns

Нам необходимо убрать nan из колонок. Не буду использовать удаление пустых строк или столбцов, поскольку это может сильно ударить по точности модели (видим, что есть колонки, где очень много nan).

Буду использовать импюютацию константой для категориальных переменных, а для числовых - медианной.

```
In [48]: for column in data.select_dtypes(include = ['int64', 'float64']).columns.tolist():
    data[column] = data[column].fillna(data[column].median())
for column in data.select_dtypes(include = ['category']).columns.tolist():
    data[column] = data[column].fillna(data[column].describe(include=['category'])['top'])
data.head()
```

Out[48]:

		Unnamed: 0	Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Special	...	Composure	Marking	StandingTackle	Slidin
0	0	L. Messi	31	Argentina	94	94		FC Barcelona	€110.5M	€565K	2202	...	96.0	33.0	28.0	
1	1	Cristiano Ronaldo	33	Portugal	94	94		Juventus	€77M	€405K	2228	...	95.0	28.0	31.0	
2	2	Neymar Jr	26	Brazil	92	93		Paris Saint- Germain	€118.5M	€290K	2143	...	94.0	27.0	24.0	
3	3	De Gea	27	Spain	91	93		Manchester United	€72M	€260K	1471	...	68.0	15.0	21.0	
4	4	K. De Bruyne	27	Belgium	91	92		Manchester City	€102M	€355K	2281	...	88.0	68.0	58.0	

5 rows × 85 columns

Закодируем категориальные признаки числовыми значениями

```
In [49]: le = LabelEncoder()
```

```
def CostCoding(x):
    if x[-1]=='K':
        x = float(x.lstrip('€').rstrip('KM'))/1000
    else:
        x = float(x.lstrip('€').rstrip('KM'))
    return x

data['Release Clause'] = data['Release Clause'].apply(lambda x: CostCoding(x))
data['Release Clause'] = data['Release Clause'].astype('float64')
for column in data.select_dtypes(include = ['category']).columns.tolist():
    le.fit(data[column])
    data[column] = le.transform(data[column])
#кодирование категориальных признаков
data.head(18207)
```

Out[49]:

	Unnamed: 0	Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Special	...	Composure	Marking	StandingTackle	SlidingTackle
0	0	9632	31	6	94	94	212	16	94	2202	...	96.0	33.0	28.0	26.0
1	1	3153	33	123	94	94	326	195	74	2228	...	95.0	28.0	31.0	23.0
2	2	12508	26	20	92	93	435	18	55	2143	...	94.0	27.0	24.0	33.0
3	3	4136	27	139	91	93	375	190	49	1471	...	68.0	15.0	21.0	13.0
4	4	8617	27	13	91	92	374	12	66	2281	...	88.0	68.0	58.0	51.0
5	5	4419	27	13	91	91	134	213	64	2142	...	91.0	34.0	27.0	22.0
6	6	9640	32	35	91	91	470	182	77	2280	...	84.0	60.0	76.0	73.0
7	7	9841	31	158	91	91	212	201	81	2346	...	85.0	62.0	45.0	38.0
8	8	15417	32	139	91	91	470	154	70	2201	...	82.0	87.0	92.0	91.0
9	9	7775	25	136	90	93	61	183	137	1331	...	70.0	27.0	12.0	18.0
10	10	13855	29	122	90	90	214	195	33	2152	...	86.0	34.0	42.0	19.0
11	11	15794	28	59	90	90	470	193	66	2190	...	85.0	72.0	79.0	69.0
12	12	3476	32	158	90	90	61	140	8	1946	...	82.0	90.0	89.0	89.0
13	13	4129	32	139	90	90	374	172	53	2115	...	93.0	59.0	53.0	29.0
14	14	12232	27	55	89	90	134	177	38	2189	...	85.0	90.0	91.0	85.0
15	15	12962	24	6	89	94	326	207	33	2092	...	84.0	23.0	20.0	20.0
16	16	6232	24	46	89	91	583	204	33	2165	...	89.0	56.0	36.0	38.0
17	17	580	27	55	89	90	61	196	14	2246	...	87.0	59.0	47.0	48.0
18	18	11781	26	59	89	92	212	165	43	1328	...	69.0	25.0	13.0	10.0
19	19	15618	26	13	89	90	470	157	43	1311	...	66.0	20.0	18.0	16.0
20	20	15408	29	139	89	89	212	153	60	2065	...	90.0	90.0	86.0	80.0
21	21	4327	31	158	89	89	435	172	32	2161	...	82.0	52.0	45.0	39.0
22	22	11197	32	59	89	89	214	116	10	1473	...	70.0	17.0	10.0	11.0
23	23	14390	30	6	89	89	374	178	58	2107	...	90.0	30.0	20.0	12.0
24	24	5571	33	78	89	89	326	78	36	1841	...	84.0	93.0	93.0	90.0

Unnamed: 0		Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Special	...	Composure	Marking	Standing Tackle	Sliding Tackle
25	25	8831	19	55	88	95	435	202	1	2118	...	86.0	34.0	34.0	32.0
26	26	11459	26	44	88	89	363	184	47	2146	...	91.0	38.0	43.0	41.0
27	27	3054	26	20	88	90	470	166	53	2170	...	84.0	88.0	90.0	87.0
28	28	7911	26	31	88	89	214	184	60	2171	...	87.0	52.0	41.0	44.0
29	29	9511	27	78	88	88	398	176	20	2017	...	83.0	51.0	24.0	22.0
...
18177	18177	14022	18	126	48	69	80	188	31	1178	...	49.0	18.0	16.0	11.0
18178	18178	9896	18	144	48	65	177	151	31	738	...	28.0	16.0	11.0	10.0
18179	18179	8180	17	46	48	64	407	151	31	1166	...	37.0	42.0	51.0	49.0
18180	18180	10832	22	131	48	58	556	130	31	987	...	28.0	12.0	15.0	16.0
18181	18181	2677	17	126	48	66	92	151	31	1296	...	38.0	43.0	49.0	45.0
18182	18182	16790	18	31	48	65	63	151	31	1311	...	38.0	44.0	42.0	46.0
18183	18183	8930	44	46	48	48	122	0	31	774	...	56.0	15.0	15.0	13.0
18184	18184	3527	18	46	48	55	362	130	31	1368	...	42.0	47.0	49.0	53.0
18185	18185	4659	19	126	48	59	188	151	31	1315	...	43.0	39.0	39.0	48.0
18186	18186	17107	20	30	47	64	73	171	31	1389	...	39.0	53.0	41.0	51.0
18187	18187	2462	19	59	47	59	547	130	31	1366	...	47.0	40.0	42.0	42.0
18188	18188	9342	17	161	47	62	406	171	31	1297	...	46.0	33.0	38.0	41.0
18189	18189	704	18	59	47	61	547	171	31	1290	...	37.0	28.0	15.0	22.0
18190	18190	9905	18	46	47	67	122	171	31	1285	...	46.0	35.0	44.0	47.0
18191	18191	7758	18	46	47	65	122	151	31	1250	...	36.0	45.0	42.0	46.0
18192	18192	15216	18	46	47	64	122	151	31	1325	...	38.0	41.0	41.0	44.0
18193	18193	12163	18	29	47	64	599	151	31	1191	...	32.0	41.0	48.0	48.0
18194	18194	7685	18	78	47	65	353	151	31	731	...	23.0	6.0	10.0	11.0
18195	18195	14711	18	126	47	67	629	171	31	1325	...	41.0	44.0	37.0	48.0

Unnamed: 0		Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Special	...	Composure	Marking	Standing Tackle	Sliding Tackle
18196	18196	8667	19	81	47	61	327	171	31	1333	...	35.0	41.0	44.0	54.0
18197	18197	3521	18	126	47	61	160	171	31	1362	...	52.0	41.0	47.0	38.0
18198	18198	7566	18	46	47	70	102	171	31	792	...	34.0	15.0	11.0	13.0
18199	18199	10283	18	25	47	69	607	188	31	1303	...	40.0	48.0	49.0	49.0
18200	18200	8198	18	131	47	62	571	171	31	1203	...	50.0	15.0	17.0	14.0
18201	18201	3998	18	126	47	68	629	171	31	1098	...	43.0	44.0	47.0	53.0
18202	18202	7580	19	46	47	65	165	171	31	1307	...	45.0	40.0	48.0	47.0
18203	18203	12101	19	144	47	63	587	171	31	1098	...	42.0	22.0	15.0	19.0
18204	18204	2133	16	46	47	67	122	171	31	1189	...	41.0	32.0	13.0	11.0
18205	18205	3997	17	46	47	66	586	171	31	1228	...	46.0	20.0	25.0	27.0
18206	18206	5807	16	46	46	66	586	171	31	1321	...	43.0	40.0	43.0	50.0

18207 rows × 85 columns

Отмасштабирую данные от 0 до 1

```
In [50]: MMSc = MinMaxScaler()
for column in data.columns.tolist():
    data[column] = MMSc.fit_transform(data[[column]])
```

```
In [51]: data.head()
```

Out[51]:

	Unnamed: 0	Name	Age	Nationality	Overall	Potential	Club	Value	Wage	Special	...	Composure	Marking	StandingTac
0	0.000000	0.560228	0.517241	0.036810	1.000000	0.978723	0.326154	0.074074	0.657343	0.910836	...	1.000000	0.329670	0.2857
1	0.000055	0.183389	0.586207	0.754601	1.000000	0.978723	0.501538	0.902778	0.517483	0.926935	...	0.989247	0.274725	0.3186
2	0.000110	0.727505	0.344828	0.122699	0.958333	0.957447	0.669231	0.083333	0.384615	0.874303	...	0.978495	0.263736	0.2417
3	0.000165	0.240563	0.379310	0.852761	0.937500	0.957447	0.576923	0.879630	0.342657	0.458204	...	0.698925	0.131868	0.2087
4	0.000220	0.501192	0.379310	0.079755	0.937500	0.936170	0.575385	0.055556	0.461538	0.959752	...	0.913978	0.714286	0.6153

5 rows × 85 columns

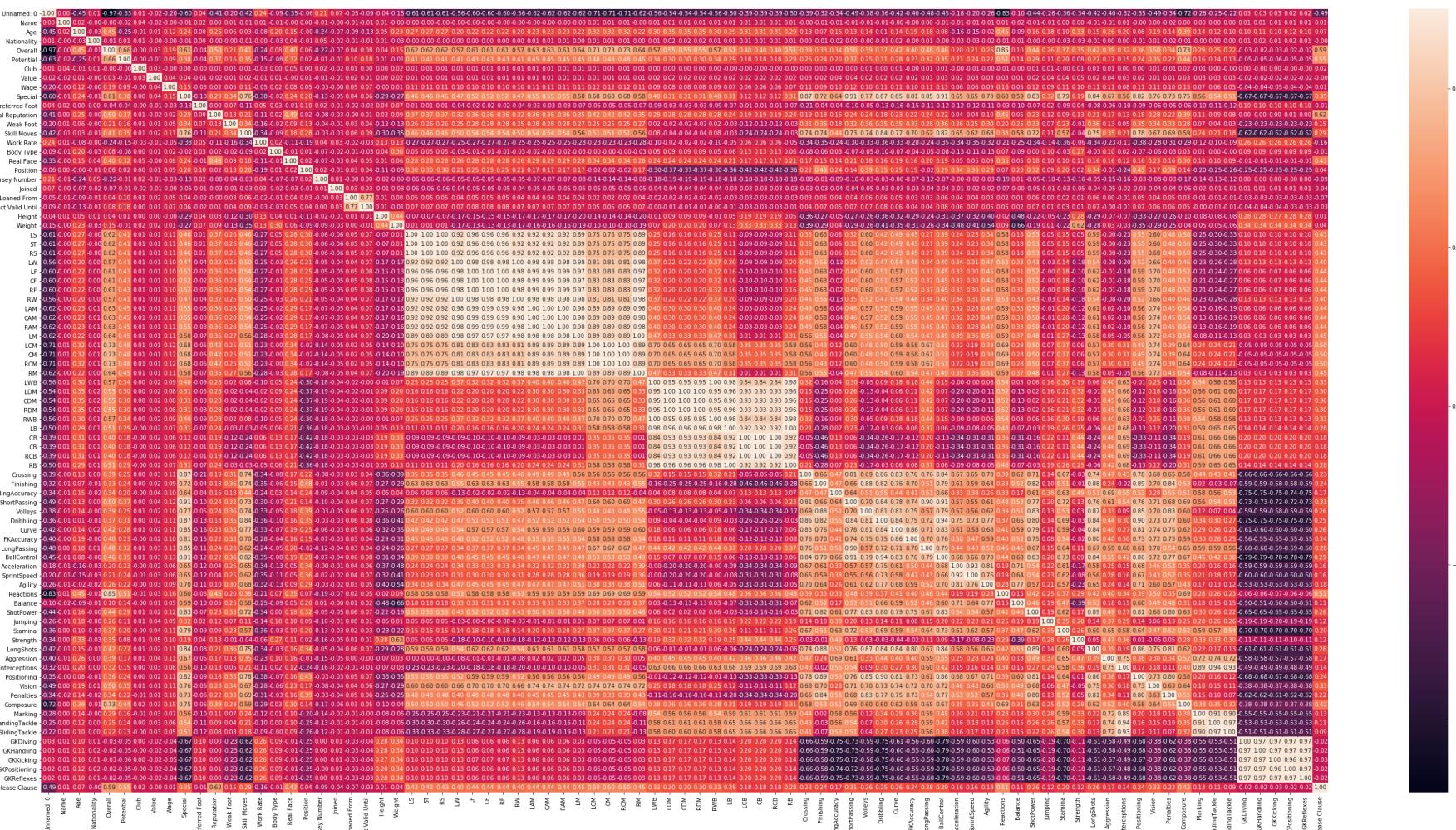
На текущий момент в наше выборке отсутствуют пропуски и все категориальные признаки закодированы целочисленными значениями. Можем приступать к анализу данных.

Анализ данных

Анализ данных начнем с построения матрицы корреляций:

```
In [52]: fig, ax = plt.subplots(figsize=(50,25))
sns.heatmap(data.corr(method='pearson'), ax=ax, annot=True, fmt='.2f')
```

Out[52]: <matplotlib.axes._subplots.AxesSubplot at 0x24bda768160>



Heh, mdaaaa

Удалю все фичи, корреляция которых с Целевой фичей будет примерно меньше 0.25 по модулю

```
In [53]: lgbm_regressor = lightgbm.LGBMRegressor().fit(data.loc[:, data.columns != 'Release Clause'], data['Release Clause'])
list_of_importances = list(zip(data.loc[:, data.columns != 'Release Clause'].columns.tolist(),
                                lgbm_regressor.feature_importances_))
list_of_importances = sorted(list_of_importances, key= lambda x: x[1], reverse= True) # список фич, отсортированных по
# важности
important_features = [x[0] for x in list_of_importances if x[1] >= 40]
important_features.extend(['Release Clause'])
data = data[important_features]
list_of_importances
```

Out[53]: [('Value', 310),
('Potential', 240),
('Unnamed: 0', 205),
('Age', 165),
('Wage', 147),
('Contract Valid Until', 141),
('Nationality', 125),
('Club', 100),
('Overall', 99),
('LF', 80),
('LS', 64),
('Reactions', 55),
('LCM', 50),
('Volleys', 48),
('Finishing', 46),
('Dribbling', 44),
('BallControl', 43),
('Composure', 43),
('LAM', 37),
('LongPassing', 37),
('Special', 35),
('LM', 34),
('GKDiving', 34),
('HeadingAccuracy', 32),
('Stamina', 32),
('LongShots', 30),
('Vision', 30),
('GKPositioning', 29),
('GKReflexes', 29),
('Real Face', 28),
('Crossing', 28),
('Curve', 26),
('Positioning', 26),
('Name', 25),
('Joined', 24),
('Jumping', 24),
('LCB', 23),
('FKAccuracy', 23),
('StandingTackle', 23),
('ShortPassing', 22),
('Agility', 22),

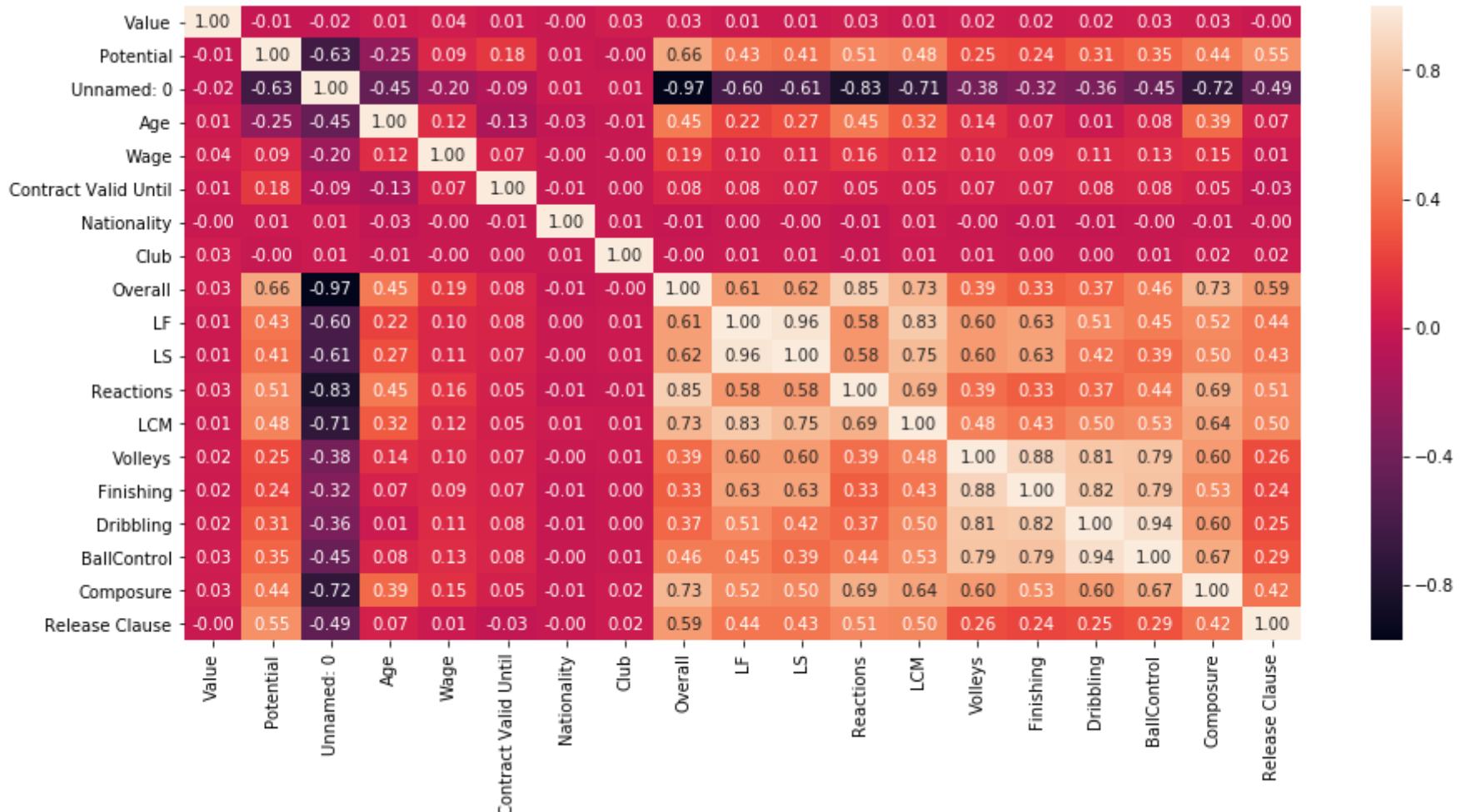
```
('LW', 20),  
('LDM', 19),  
('Interceptions', 19),  
('Marking', 19),  
('Jersey Number', 17),  
('LWB', 17),  
('International Reputation', 16),  
('Loaned From', 16),  
('GKHandling', 16),  
('SlidingTackle', 15),  
('Position', 14),  
('Height', 14),  
('Weight', 14),  
('ShotPower', 14),  
('SprintSpeed', 13),  
('Strength', 13),  
('Aggression', 13),  
('Acceleration', 12),  
('Penalties', 12),  
('GKKicking', 12),  
('LB', 10),  
('Work Rate', 7),  
('Balance', 7),  
('Preferred Foot', 6),  
('Weak Foot', 5),  
('Skill Moves', 1),  
('Body Type', 1),  
('ST', 0),  
('RS', 0),  
('CF', 0),  
('RF', 0),  
('RW', 0),  
('CAM', 0),  
('RAM', 0),  
('CM', 0),  
('RCM', 0),  
('RM', 0),  
('CDM', 0),  
('RDM', 0),  
('RWB', 0),  
('CB', 0),
```

```
('RCB', 0),  
('RB', 0)]
```

Теперь у нас есть только подходящие данные для анализа. Еще раз посмотрим на матрицу корреляций

```
In [55]: fig, ax = plt.subplots(figsize=(15, 7))  
sns.heatmap(data.corr(method='pearson'), ax=ax, annot=True, fmt='.2f')
```

```
Out[55]: <matplotlib.axes._subplots.AxesSubplot at 0x24be56edda0>
```



Избавлюсь ещё от нескольких признаков, которые не заметила lgbm, но заметила Корр.Матрица

```
In [56]: data = data.drop(['Value', 'Wage', 'Contract Valid Until', 'Nationality', 'Club', 'Unnamed: 0', 'Age'], axis=1)
```

```
In [57]: fig, ax = plt.subplots(figsize=(10, 5))
sns.heatmap(data.corr(method='pearson'), ax=ax, annot=True, fmt='.2f')
```

```
Out[57]: <matplotlib.axes._subplots.AxesSubplot at 0x24be5b71e48>
```

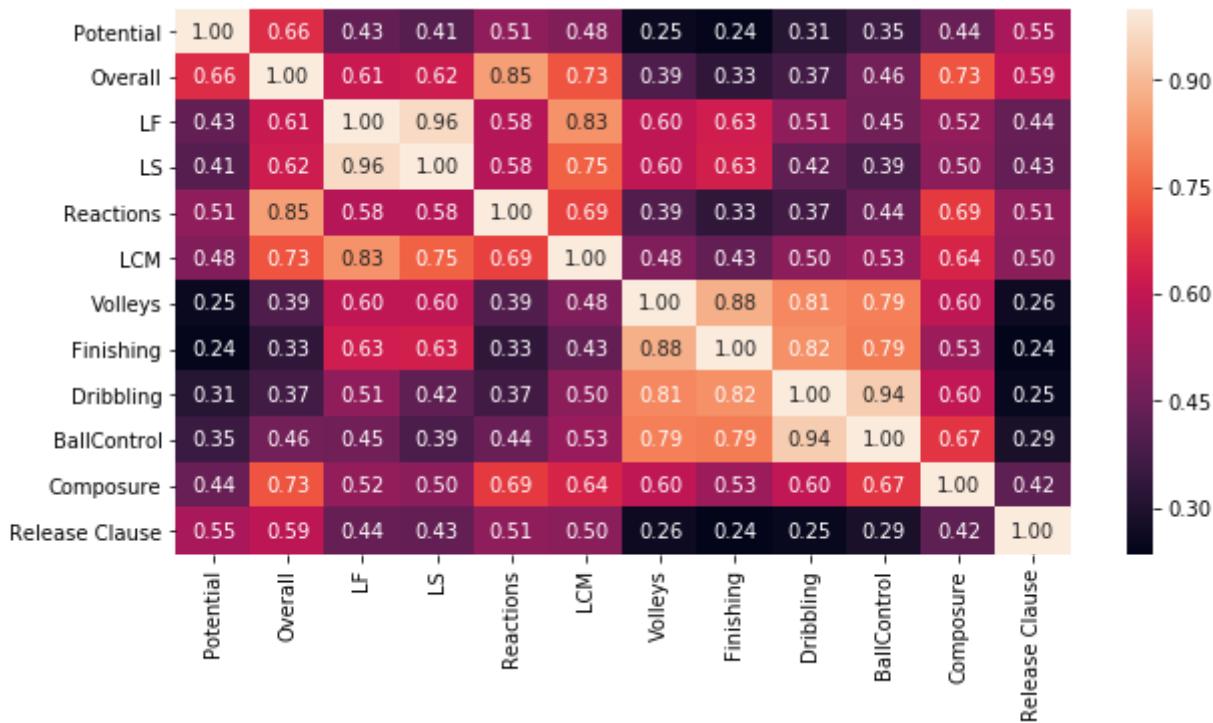
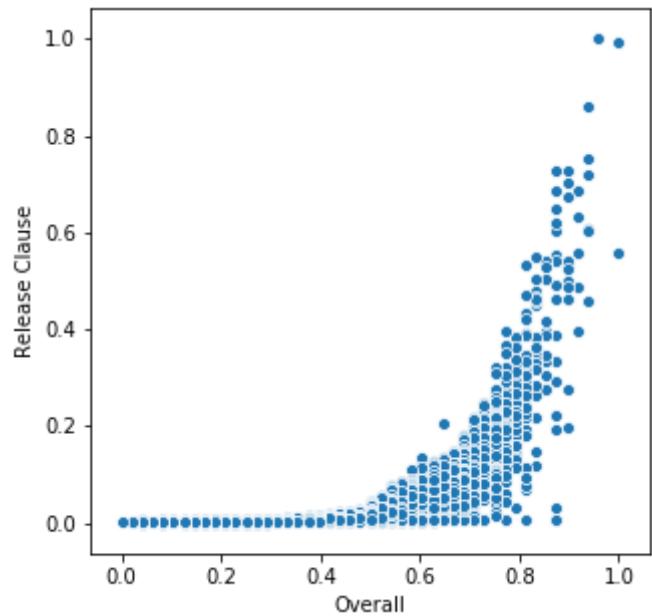


Диаграмма рассеиваний показывает зависимость двух признаков:

```
In [58]: fig, ax = plt.subplots(figsize=(5,5))
sns.scatterplot(ax=ax, x='Overall', y='Release Clause', data=data)
```

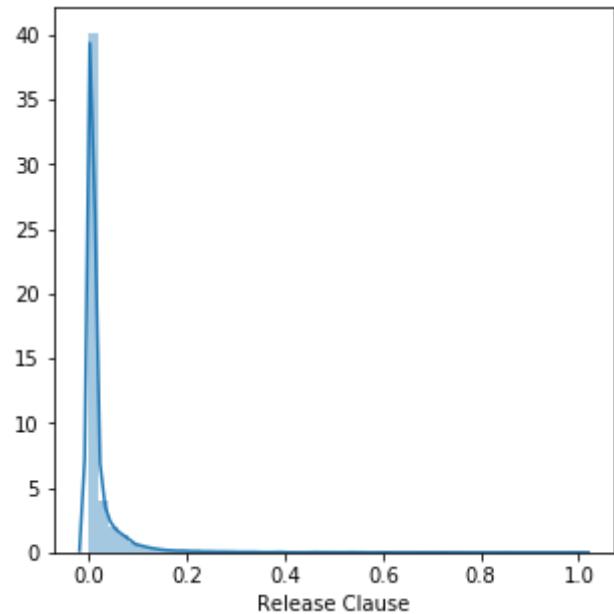
```
Out[58]: <matplotlib.axes._subplots.AxesSubplot at 0x24be42f9c50>
```



Видна экспоненциальная зависимость.

```
In [59]: #Гистограмма Позволяет оценить плотность распределения данных  
fig, ax = plt.subplots(figsize=(5,5))  
sns.distplot(data['Release Clause'])
```

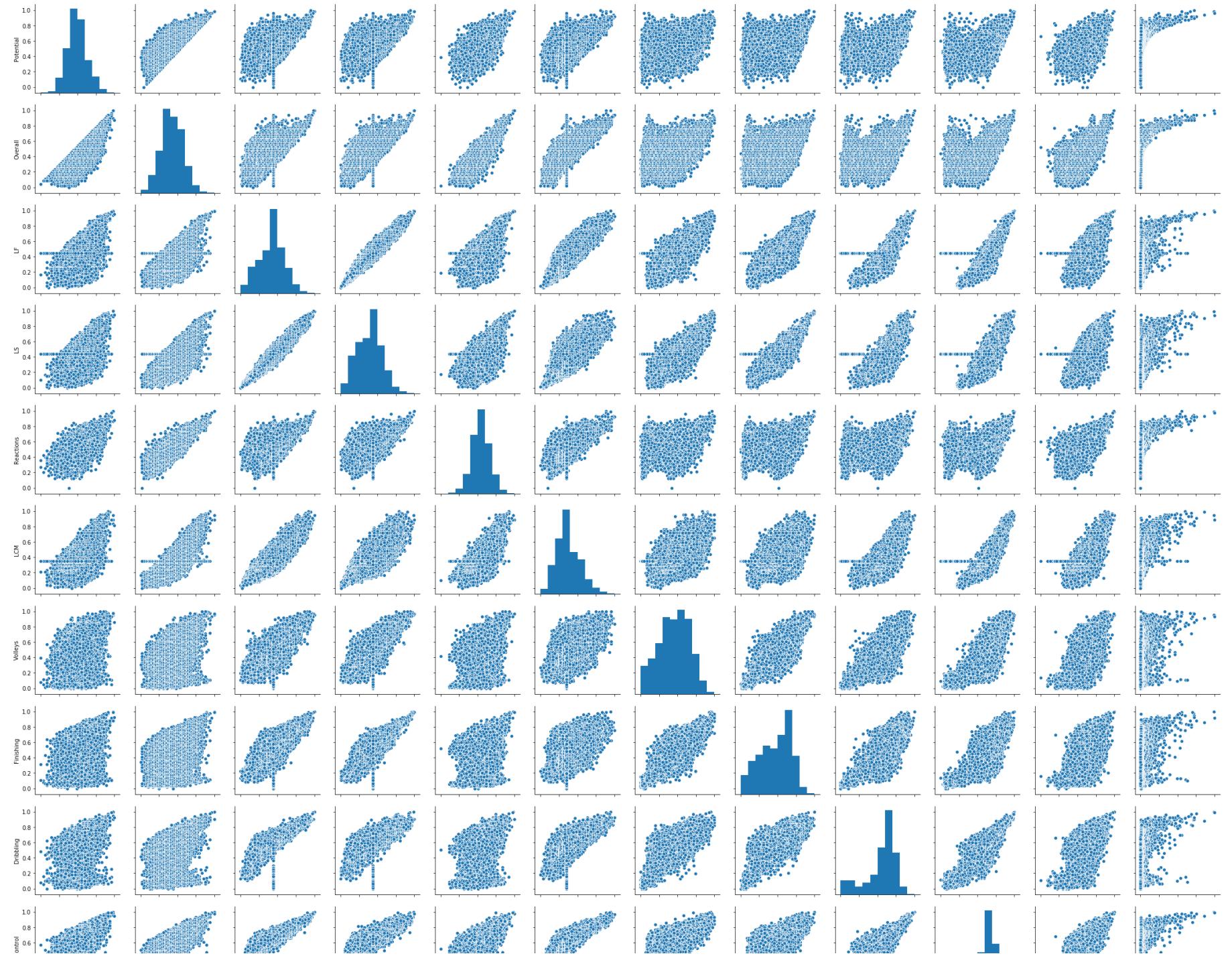
```
Out[59]: <matplotlib.axes._subplots.AxesSubplot at 0x24be433f208>
```

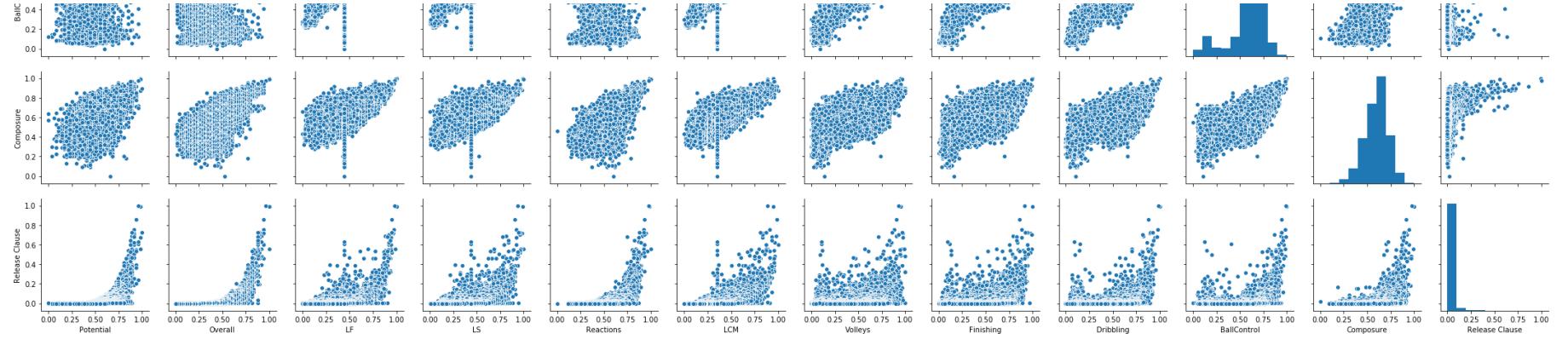


Посмотрим, как все признаки зависят между собой

```
In [60]: sns.pairplot(data)
```

Out[60]: <seaborn.axisgrid.PairGrid at 0x24be4377cc0>





Опять заметно, что некоторые данные находятся в нелинейной зависимости

Разделение выборки

Для начала разделим целевой признак от остальных:

```
In [61]: data_x = data.loc[:, data.columns != 'Release Clause']
data_y = data[['Release Clause']]
```

И теперь разделим на тренировочную выборку и тестовую, в тренировочной оставим 70% от всех данных

```
In [67]: data_X_train, data_X_test, data_y_train, data_y_test = train_test_split(
    data_x, data_y, test_size=0.33, random_state=1)
data_X_train.shape, data_X_test.shape
```

```
Out[67]: ((12198, 11), (6009, 11))
```

Метод ближайших соседей

Начнем с одного из самых простых методов.

Сначала попробуем обучать на основе двух ближайших соседей

```
In [217]: KNN1 = KNeighborsRegressor(n_neighbors=2)
KNN1.fit(data_X_train, data_y_train)
target_KNN1 = KNN1.predict(data_X_test)
target_KNN1
```

```
Out[217]: array([[0.00042089],
 [0.01726096],
 [0.00254947],
 ...,
 [0.00268976],
 [0.03984006],
 [0.00045816]])
```

```
In [218]: print_stat(data_y_test, target_KNN1)
#средняя абсолютная ошибка при 2 соседях
#средняя квадратичная ошибка при 2 соседях
#медианная абсолютная ошибка при 2 соседях
```

```
0.00631380651923172
0.00034694691584392863
0.001315287587631035
```

Теперь с помощью кросс-валидации подберем гиперпараметр:

```
In [219]: n_range = np.array(range(2,51,4))
tuned_parameters = [{n_neighbors: n_range}]
tuned_parameters
```

```
Out[219]: [{n_neighbors: array([ 2,  6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50])}]
```

```
In [220]: clf_gs = GridSearchCV(KNeighborsRegressor(), tuned_parameters, cv=5, scoring='neg_mean_absolute_error')
clf_gs.fit(data_X_train, data_y_train)
```

```
Out[220]: GridSearchCV(cv=5, error_score='raise-deprecating',
                       estimator=KNeighborsRegressor(algorithm='auto', leaf_size=30,
                                                     metric='minkowski',
                                                     metric_params=None, n_jobs=None,
                                                     n_neighbors=5, p=2,
                                                     weights='uniform'),
                       iid='warn', n_jobs=None,
                       param_grid=[{'n_neighbors': array([ 2,  6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50])}],
                       pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
                       scoring='neg_mean_absolute_error', verbose=0)
```

```
In [221]: clf_gs.best_params_
```

```
Out[221]: {'n_neighbors': 6}
```

```
In [222]: #6 ближайших соседей
KNN2 = KNeighborsRegressor(n_neighbors=6)
KNN2.fit(data_X_train, data_y_train)
target_KNN2 = KNN2.predict(data_X_test)
target_KNN2
```

```
Out[222]: array([[0.00061161],
                  [0.02244904],
                  [0.00206208],
                  ...,
                  [0.00362289],
                  [0.04524735],
                  [0.00044939]])
```

```
In [223]: print_stat(data_y_test, target_KNN_2)
```

```
#Было:  
#0.00631380651923172  
#0.00034694691584392863  
#0.001315287587631035
```

```
#Стало лучше:
```

```
0.005551406067761183  
0.0003131662338322637  
0.0011041108582836084
```

Средняя абсолютная и квадратичная ошибка стали намного меньше

Линейная модель

Некоторые данные находятся в линейной зависимости, поэтому попробуем линейную модель

```
In [121]: # Аналитическое вычисление коэффициентов регрессии  
def analytic_regr_coef(x_array : np.ndarray,  
                      y_array : np.ndarray) -> Tuple[float, float]:  
    x_mean = np.mean(x_array)  
    y_mean = np.mean(y_array)  
    var1 = np.sum([(x-x_mean)**2 for x in x_array])  
    cov1 = np.sum([(x-x_mean)*(y-y_mean) for x, y in zip(x_array, y_array)])  
    b1 = cov1 / var1  
    b0 = y_mean - b1*x_mean  
    return b0, b1
```

Для начала найдем коэффициенты линейной зависимости и наглядно убедимся, насколько наша зависимость похожа на линейную

```
In [122]: x_array = data[['LS']]
y_array = data[['Release Clause']]
```

```
In [123]: df1 = pd.DataFrame(x_array)
df2 = pd.DataFrame(y_array)
```

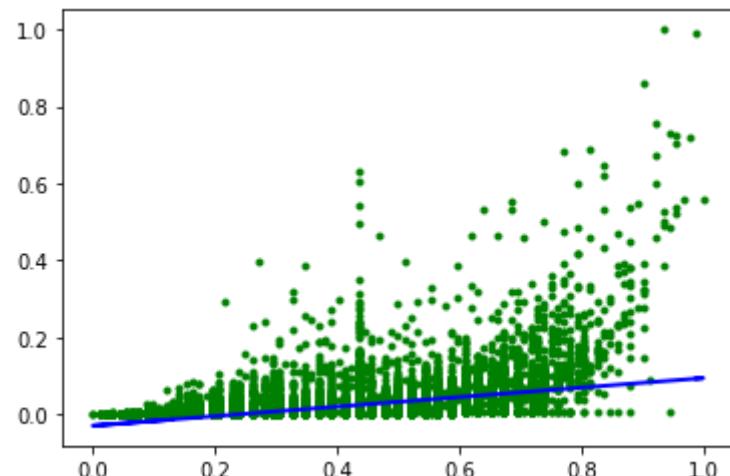
```
In [124]: b0, b1 = analytic_regr_coef(df1.values, df2.values)
b0, b1
```

```
Out[124]: (-0.02946060733603696, 0.12373310043946867)
```

```
In [125]: # Вычисление значений у на основе x для регрессии
def y_regr(x_array : np.ndarray, b0: float, b1: float) -> np.ndarray:
    res = [b1*x+b0 for x in x_array]
    return res
```

```
In [126]: y_array_regr = y_regr(df1.values, b0, b1)
```

```
In [127]: plt.plot(x_array, y_array, 'g.')
plt.plot(x_array, y_array_regr, 'b', linewidth=2.0)
plt.show()
```

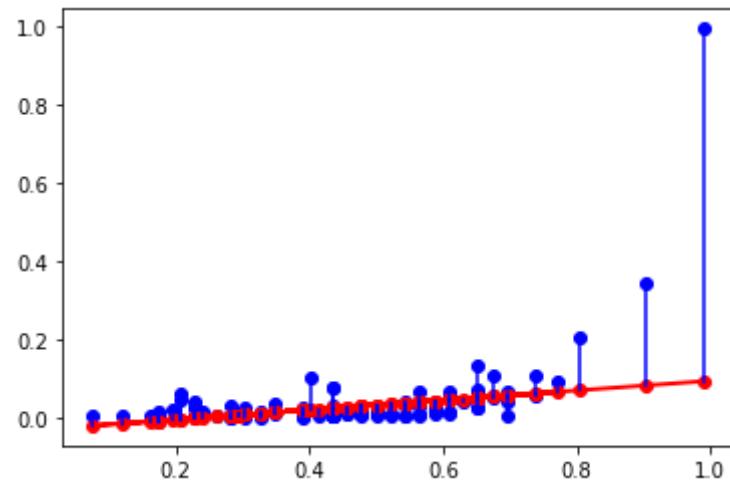


Можно посмотреть, насколько данные близко к линии. Синими отрезками показаны ошибки между истинными и предсказанными значениями

```
In [128]: plt.plot(df1.values[0:10000:100], df2.values[0:10000:100], 'bo')
plt.plot(df1.values[0:10000:100], y_array_regr[0:10000:100], '-ro', linewidth=2.0)

for i in range(len(x_array[0:10000:100])):
    x1 = df1.values[0:10000:100][i]
    y1 = df2.values[0:10000:100][i]
    y2 = y_array_regr[0:10000:100][i]
    plt.plot([x1,x1],[y1,y2], 'b-')

plt.show()
```



Попробуем обучить модель и предсказать значения:

```
In [129]: LR1 = LinearRegression().fit(data_X_train, data_y_train)
```

```
In [130]: target_LR1 = LR1.predict(data_X_test)
```

```
In [131]: print_stat(data_y_test, target_LR1)
```

```
0.01841404221130518  
0.0012439963515848984  
0.013742501790125167
```

До подбора гиперпараметров наша модель показывает плохой результат

```
In [132]: model = LinearRegression()  
parameters = {'fit_intercept':[True, False], 'normalize':[True, False], 'copy_X':[True, False]}  
grid = GridSearchCV(model, parameters, cv=5)  
grid.fit(data_X_train, data_y_train)
```

```
Out[132]: GridSearchCV(cv=5, error_score='raise-deprecating',  
estimator=LinearRegression(copy_X=True, fit_intercept=True,  
n_jobs=None, normalize=False),  
iid='warn', n_jobs=None,  
param_grid={'copy_X': [True, False],  
'fit_intercept': [True, False],  
'normalize': [True, False]},  
pre_dispatch='2*n_jobs', refit=True, return_train_score=False,  
scoring=None, verbose=0)
```

```
In [133]: grid.best_params_
```

```
Out[133]: {'copy_X': True, 'fit_intercept': True, 'normalize': True}
```

Кросс-валидация выбрала лучшими параметры. Подставим их.

```
In [136]: LR2 = LinearRegression(copy_X = True, fit_intercept = False, normalize = True).fit(data_X_train, data_y_train)
```

```
In [137]: target_LR2 = LR2.predict(data_X_test)
```

```
In [139]: print_stat(data_y_test, target_LR2)
```

```
#Было:  
#0.01841404221130518  
#0.0012439963515848984  
#0.013742501790125167
```

```
#практически ничего не поменялось, значит, линейная модель явно не подходит нашей ПО:
```

```
0.01913900992439537  
0.0013765834253408437  
0.013694985012512713
```

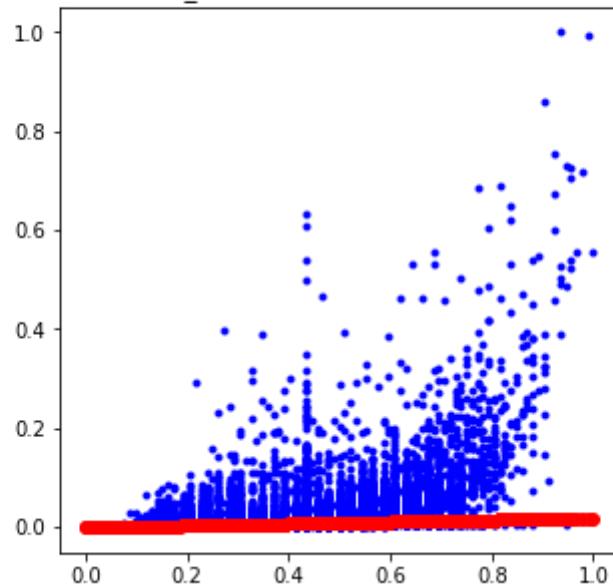
Метод опорных векторов

```
In [160]: xx = df1.values  
yy = df2.values  
def plot_regr(clf):  
    title = clf.__repr__  
    clf.fit(xx.reshape(-1, 1), yy)  
    y_pred = clf.predict(xx.reshape(-1, 1))  
    fig, ax = plt.subplots(figsize=(5,5))  
    ax.set_title(title)  
    ax.plot(xx, yy, 'b.')  
    ax.plot(xx, y_pred, 'ro')  
    plt.show()
```

```
In [161]: plot_regr(LinearSVR(C=1.0, max_iter=10000))
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)
```

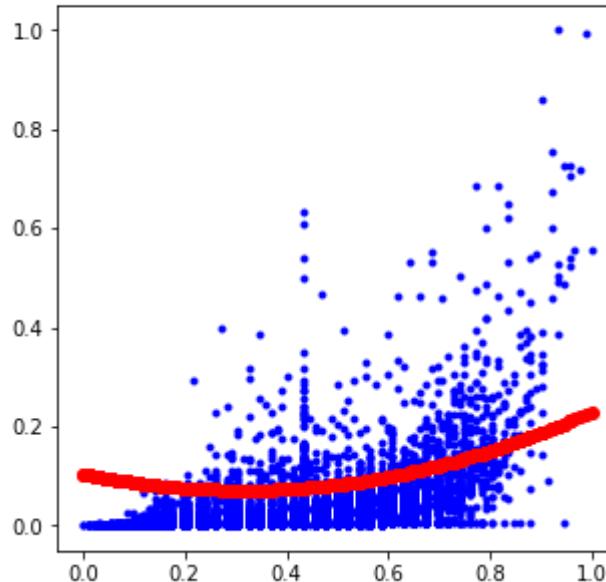
```
<bound method BaseEstimator.__repr__ of LinearSVR(C=1.0, dual=True, epsilon=0.0, fit_intercept=True,  
intercept_scaling=1.0, loss='epsilon_insensitive', max_iter=10000,  
random_state=None, tol=0.0001, verbose=0)>
```



```
In [162]: plot_regr(SVR(kernel='rbf', gamma=0.2, C=1.0))
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)
```

```
<bound method BaseEstimator.__repr__ of SVR(C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma=0.2,  
kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)>
```



```
In [163]: SVR1 = SVR().fit(data_X_train, data_y_train)
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\svm\base.py:193: FutureWarning: The default value of gamma will c  
hange from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto'  
or 'scale' to avoid this warning.  
"avoid this warning.", FutureWarning)
```

```
In [164]: target_SVR1 = SVR1.predict(data_X_test)
```

```
In [165]: print_stat(data_y_test, target_SVR1)
```

```
0.03918261227310407  
0.0022940984952448448  
0.03682178501495414
```

```
In [166]: param_grid = {'C':[1,10,100,1000], 'gamma':[1,0.1,0.001,0.0001], 'kernel':['linear','rbf']}
grid = GridSearchCV(SVR(),param_grid,refit = True, verbose=2)
grid.fit(data_X_train, data_y_train)
```

```
c:\program files\python37\lib\site-packages\sklearn\model_selection\_split.py:1978: FutureWarning: The default value  
of cv will change from 3 to 5 in version 0.22. Specify it explicitly to silence this warning.  
    warnings.warn(CV_WARNING, FutureWarning)  
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
Fitting 3 folds for each of 32 candidates, totalling 96 fits  
[CV] C=1, gamma=1, kernel=linear .....  
[CV] ..... C=1, gamma=1, kernel=linear, total= 0.1s  
[CV] C=1, gamma=1, kernel=linear .....  
  
[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.0s remaining: 0.0s  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=1, gamma=1, kernel=linear, total= 0.1s  
[CV] C=1, gamma=1, kernel=linear .....  
[CV] ..... C=1, gamma=1, kernel=linear, total= 0.2s  
[CV] C=1, gamma=1, kernel=rbf .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)
```

```
[CV] ..... C=1, gamma=1, kernel=rbf, total= 0.1s
[CV] C=1, gamma=1, kernel=rbf .....
[CV] ..... C=1, gamma=1, kernel=rbf, total= 0.0s
[CV] C=1, gamma=1, kernel=rbf .....
[CV] ..... C=1, gamma=1, kernel=rbf, total= 0.0s
[CV] C=1, gamma=0.1, kernel=linear .....
[CV] ..... C=1, gamma=0.1, kernel=linear, total= 0.1s
[CV] C=1, gamma=0.1, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[CV] ..... C=1, gamma=0.1, kernel=linear, total= 0.2s
[CV] C=1, gamma=0.1, kernel=linear .....
[CV] ..... C=1, gamma=0.1, kernel=linear, total= 0.2s
[CV] C=1, gamma=0.1, kernel=rbf .....
[CV] ..... C=1, gamma=0.1, kernel=rbf, total= 0.0s
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[CV] C=1, gamma=0.1, kernel=rbf .....
[CV] ..... C=1, gamma=0.1, kernel=rbf, total= 0.1s
[CV] C=1, gamma=0.1, kernel=rbf .....
[CV] ..... C=1, gamma=0.1, kernel=rbf, total= 0.1s
[CV] C=1, gamma=0.001, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[CV] ..... C=1, gamma=0.001, kernel=linear, total= 0.2s
[CV] C=1, gamma=0.001, kernel=linear .....
[CV] ..... C=1, gamma=0.001, kernel=linear, total= 0.1s
[CV] C=1, gamma=0.001, kernel=linear .....

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)

[CV] ..... C=1, gamma=0.001, kernel=linear, total= 0.2s
[CV] C=1, gamma=0.001, kernel=rbf .....
[CV] ..... C=1, gamma=0.001, kernel=rbf, total= 0.1s
[CV] C=1, gamma=0.001, kernel=rbf .....

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)

[CV] ..... C=1, gamma=0.001, kernel=rbf, total= 0.1s
[CV] C=1, gamma=0.001, kernel=rbf .....
[CV] ..... C=1, gamma=0.001, kernel=rbf, total= 0.1s
[CV] C=1, gamma=0.0001, kernel=linear .....

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)

[CV] ..... C=1, gamma=0.0001, kernel=linear, total= 0.2s
[CV] C=1, gamma=0.0001, kernel=linear .....
[CV] ..... C=1, gamma=0.0001, kernel=linear, total= 0.1s
[CV] C=1, gamma=0.0001, kernel=linear .....
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=1, gamma=0.0001, kernel=linear, total= 0.2s  
[CV] C=1, gamma=0.0001, kernel=rbf .....  
[CV] ..... C=1, gamma=0.0001, kernel=rbf, total= 0.1s  
[CV] C=1, gamma=0.0001, kernel=rbf .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=1, gamma=0.0001, kernel=rbf, total= 0.2s  
[CV] C=1, gamma=0.0001, kernel=rbf .....  
[CV] ..... C=1, gamma=0.0001, kernel=rbf, total= 0.2s  
[CV] C=10, gamma=1, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=10, gamma=1, kernel=linear, total= 0.5s  
[CV] C=10, gamma=1, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=10, gamma=1, kernel=linear, total= 0.3s  
[CV] C=10, gamma=1, kernel=linear .....
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=10, gamma=1, kernel=linear, total= 0.4s  
[CV] C=10, gamma=1, kernel=rbf .....  
[CV] ..... C=10, gamma=1, kernel=rbf, total= 0.1s  
[CV] C=10, gamma=1, kernel=rbf .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=10, gamma=1, kernel=rbf, total= 0.2s  
[CV] C=10, gamma=1, kernel=rbf .....  
[CV] ..... C=10, gamma=1, kernel=rbf, total= 0.1s  
[CV] C=10, gamma=0.1, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=10, gamma=0.1, kernel=linear, total= 0.3s  
[CV] C=10, gamma=0.1, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=10, gamma=0.1, kernel=linear, total= 0.4s  
[CV] C=10, gamma=0.1, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)
```

```
[CV] ..... C=10, gamma=0.1, kernel=linear, total= 0.3s
[CV] C=10, gamma=0.1, kernel=rbf .....
[CV] ..... C=10, gamma=0.1, kernel=rbf, total= 0.1s
[CV] C=10, gamma=0.1, kernel=rbf .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)
```

```
[CV] ..... C=10, gamma=0.1, kernel=rbf, total= 0.1s
[CV] C=10, gamma=0.1, kernel=rbf .....
[CV] ..... C=10, gamma=0.1, kernel=rbf, total= 0.1s
[CV] C=10, gamma=0.001, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)
```

```
[CV] ..... C=10, gamma=0.001, kernel=linear, total= 0.4s
[CV] C=10, gamma=0.001, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[CV] ..... C=10, gamma=0.001, kernel=linear, total= 0.3s
[CV] C=10, gamma=0.001, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[CV] ..... C=10, gamma=0.001, kernel=linear, total= 0.3s
[CV] C=10, gamma=0.001, kernel=rbf .....
[CV] ..... C=10, gamma=0.001, kernel=rbf, total= 0.1s
[CV] C=10, gamma=0.001, kernel=rbf .....
[CV] ..... C=10, gamma=0.001, kernel=rbf, total= 0.1s
[CV] C=10, gamma=0.001, kernel=rbf .....
[CV] ..... C=10, gamma=0.001, kernel=rbf, total= 0.1s
[CV] C=10, gamma=0.001, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[CV] ..... C=10, gamma=0.0001, kernel=linear, total= 0.4s
[CV] C=10, gamma=0.0001, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[CV] ..... C=10, gamma=0.0001, kernel=linear, total= 0.3s
[CV] C=10, gamma=0.0001, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[CV] ..... C=10, gamma=0.0001, kernel=linear, total= 0.4s
[CV] C=10, gamma=0.0001, kernel=rbf .....
```

[CV] C=10, gamma=0.0001, kernel=rbf, total= 0.1s

[CV] C=10, gamma=0.0001, kernel=rbf

[CV] C=10, gamma=0.0001, kernel=rbf, total= 0.1s

[CV] C=10, gamma=0.0001, kernel=rbf

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=10, gamma=0.0001, kernel=rbf, total= 0.1s  
[CV] C=100, gamma=1, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=100, gamma=1, kernel=linear, total= 1.1s  
[CV] C=100, gamma=1, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=100, gamma=1, kernel=linear, total= 1.2s  
[CV] C=100, gamma=1, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=100, gamma=1, kernel=linear, total= 1.4s  
[CV] C=100, gamma=1, kernel=rbf .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=100, gamma=1, kernel=rbf, total= 0.3s  
[CV] C=100, gamma=1, kernel=rbf .....  
[CV] ..... C=100, gamma=1, kernel=rbf, total= 0.2s  
[CV] C=100, gamma=1, kernel=rbf .....
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=100, gamma=1, kernel=rbf, total= 0.2s  
[CV] C=100, gamma=0.1, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=100, gamma=0.1, kernel=linear, total= 1.1s  
[CV] C=100, gamma=0.1, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=100, gamma=0.1, kernel=linear, total= 1.3s  
[CV] C=100, gamma=0.1, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
  
[CV] ..... C=100, gamma=0.1, kernel=linear, total= 1.4s  
[CV] C=100, gamma=0.1, kernel=rbf .....  
[CV] ..... C=100, gamma=0.1, kernel=rbf, total= 0.2s  
[CV] C=100, gamma=0.1, kernel=rbf .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)
```

```
[CV] ..... C=100, gamma=0.1, kernel=rbf, total= 0.2s
[CV] C=100, gamma=0.1, kernel=rbf .....
[CV] ..... C=100, gamma=0.1, kernel=rbf, total= 0.1s
[CV] C=100, gamma=0.001, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)
```

[CV] C=100, gamma=0.001, kernel=linear, total= 1.0s

```
[CV] C=100, gamma=0.001, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

[CV] C=100, gamma=0.001, kernel=linear, total= 1.3s

```
[CV] C=100, gamma=0.001, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

[CV] C=100, gamma=0.001, kernel=linear, total= 1.4s

```
[CV] C=100, gamma=0.001, kernel=rbf .....
```

[CV] C=100, gamma=0.001, kernel=rbf, total= 0.1s

```
[CV] C=100, gamma=0.001, kernel=rbf .....
```

[CV] C=100, gamma=0.001, kernel=rbf, total= 0.1s

```
[CV] C=100, gamma=0.001, kernel=rbf .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[CV] ..... C=100, gamma=0.001, kernel=rbf, total= 0.1s
[CV] C=100, gamma=0.0001, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[CV] ..... C=100, gamma=0.0001, kernel=linear, total= 1.2s
[CV] C=100, gamma=0.0001, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[CV] ..... C=100, gamma=0.0001, kernel=linear, total= 1.7s
[CV] C=100, gamma=0.0001, kernel=linear .....
```

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[CV] ..... C=100, gamma=0.0001, kernel=linear, total= 2.2s
[CV] C=100, gamma=0.0001, kernel=rbf .....
```

[CV] C=100, gamma=0.0001, kernel=rbf, total= 0.1s
[CV] C=100, gamma=0.0001, kernel=rbf

```
[CV] ..... C=100, gamma=0.0001, kernel=rbf, total= 0.1s
[CV] C=100, gamma=0.0001, kernel=rbf .....
```

[CV] C=100, gamma=0.0001, kernel=rbf, total= 0.1s
[CV] C=1000, gamma=1, kernel=linear

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[CV] ..... C=1000, gamma=1, kernel=linear, total= 8.0s
[CV] C=1000, gamma=1, kernel=linear .....
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
[CV] ..... C=1000, gamma=1, kernel=linear, total= 5.8s
[CV] C=1000, gamma=1, kernel=linear .....
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
[CV] ..... C=1000, gamma=1, kernel=linear, total= 7.7s
[CV] C=1000, gamma=1, kernel=rbf .....
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
[CV] ..... C=1000, gamma=1, kernel=rbf, total= 0.2s
[CV] C=1000, gamma=1, kernel=rbf .....
```

```
[CV] ..... C=1000, gamma=1, kernel=rbf, total= 0.1s
[CV] C=1000, gamma=1, kernel=rbf .....
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
[CV] ..... C=1000, gamma=1, kernel=rbf, total= 0.2s
[CV] C=1000, gamma=0.1, kernel=linear .....
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
[CV] ..... C=1000, gamma=0.1, kernel=linear, total= 5.6s
[CV] C=1000, gamma=0.1, kernel=linear .....
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=1000, gamma=0.1, kernel=linear, total= 5.6s  
[CV] C=1000, gamma=0.1, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=1000, gamma=0.1, kernel=linear, total= 8.4s  
[CV] C=1000, gamma=0.1, kernel=rbf .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=1000, gamma=0.1, kernel=rbf, total= 0.6s  
[CV] C=1000, gamma=0.1, kernel=rbf .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=1000, gamma=0.1, kernel=rbf, total= 0.5s  
[CV] C=1000, gamma=0.1, kernel=rbf .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=1000, gamma=0.1, kernel=rbf, total= 0.3s  
[CV] C=1000, gamma=0.001, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=1000, gamma=0.001, kernel=linear, total= 5.6s  
[CV] C=1000, gamma=0.001, kernel=linear .....
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=1000, gamma=0.001, kernel=linear, total= 6.1s  
[CV] C=1000, gamma=0.001, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=1000, gamma=0.001, kernel=linear, total= 6.9s  
[CV] C=1000, gamma=0.001, kernel=rbf .....  
[CV] ..... C=1000, gamma=0.001, kernel=rbf, total= 0.2s  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] C=1000, gamma=0.001, kernel=rbf .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=1000, gamma=0.001, kernel=rbf, total= 0.2s  
[CV] C=1000, gamma=0.001, kernel=rbf .....  
[CV] ..... C=1000, gamma=0.001, kernel=rbf, total= 0.2s  
[CV] C=1000, gamma=0.0001, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)  
  
[CV] ..... C=1000, gamma=0.0001, kernel=linear, total= 7.0s  
[CV] C=1000, gamma=0.0001, kernel=linear .....  
  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)
```

```
[CV] ..... C=1000, gamma=0.0001, kernel=linear, total= 8.9s
[CV] C=1000, gamma=0.0001, kernel=linear .....
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
[CV] ..... C=1000, gamma=0.0001, kernel=linear, total= 8.1s
[CV] C=1000, gamma=0.0001, kernel=rbf .....
[CV] ..... C=1000, gamma=0.0001, kernel=rbf, total= 0.1s
[CV] C=1000, gamma=0.0001, kernel=rbf .....
[CV] ..... C=1000, gamma=0.0001, kernel=rbf, total= 0.1s
[CV] C=1000, gamma=0.0001, kernel=rbf .....
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
[CV] ..... C=1000, gamma=0.0001, kernel=rbf, total= 0.1s
```

```
[Parallel(n_jobs=1)]: Done 96 out of 96 | elapsed: 1.9min finished
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
Out[166]: GridSearchCV(cv='warn', error_score='raise-deprecating',
                       estimator=SVR(C=1.0, cache_size=200, coef0=0.0, degree=3,
                                     epsilon=0.1, gamma='auto_deprecated', kernel='rbf',
                                     max_iter=-1, shrinking=True, tol=0.001,
                                     verbose=False),
                       iid='warn', n_jobs=None,
                       param_grid={'C': [1, 10, 100, 1000],
                                   'gamma': [1, 0.1, 0.001, 0.0001],
                                   'kernel': ['linear', 'rbf']},
                       pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
                       scoring=None, verbose=2)
```

```
In [167]: grid.best_params_
```

```
Out[167]: {'C': 1000, 'gamma': 0.001, 'kernel': 'rbf'}
```

```
In [171]: SVR2 = SVR(C=1000, gamma = 0.001, kernel = 'rbf').fit(data_X_train, data_y_train)
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
y = column_or_1d(y, warn=True)
```

```
In [172]: target_SVR2 = SVR2.predict(data_X_test)
```

```
In [174]: print_stat(data_y_test, target_SVR2)
```

```
#Было:  
#0.03918261227310407  
#0.0022940984952448448  
#0.03682178501495414
```

```
#Все стало гораздо лучше, но всё равно плохо, хуже, чем линейная модель!!!
```

```
0.03047600356857367  
0.001656156261359265  
0.025142337045255862
```

После подбора гиперпараметров метрики заметно улучшились. Посмотрим, как покажут себя более сложные модели

Ансамблевые модели

RandomForestRegressor

```
In [177]: # data_x = data.Loc[:, data.columns != 'Release Clause']
# data_y = data[['Release Clause']]
# df1 = pd.DataFrame(data_x)
# df2 = pd.DataFrame(data_y)
```

```
In [179]: RFR1 = RandomForestRegressor(random_state=1).fit(data_X_train, data_y_train)
```

```
c:\program files\python37\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning: The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
"10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\Artem\AppData\Roaming\Python\Python37\site-packages\ipykernel_launcher.py:1: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
    """Entry point for launching an IPython kernel.
```

```
In [180]: target_RFR1 = RFR1.predict(data_X_test)
```

```
In [181]: print_stat(data_y_test, target_RFR1)
```

```
0.004899101427321029
0.00027346637497344947
0.0009250957384237517
```

```
In [185]: tuned_parameters = {'n_estimators': [500, 700, 1000], 'max_depth': [None, 1, 2, 3]}
```

```
In [186]: CV_rfr = GridSearchCV(RandomForestRegressor(), param_grid=tuned_parameters, cv=5, n_jobs=-1, verbose=1)
```

```
In [187]: CV_rfr.fit(data_X_train, data_y_train)
```

Fitting 5 folds for each of 12 candidates, totalling 60 fits

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.  
[Parallel(n_jobs=-1)]: Done 42 tasks | elapsed: 4.2min  
[Parallel(n_jobs=-1)]: Done 60 out of 60 | elapsed: 5.0min finished  
c:\program files\python37\lib\site-packages\sklearn\model_selection\_search.py:714: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().  
    self.best_estimator_.fit(X, y, **fit_params)
```

```
Out[187]: GridSearchCV(cv=5, error_score='raise-deprecating',  
                      estimator=RandomForestRegressor(bootstrap=True, criterion='mse',  
                                                     max_depth=None,  
                                                     max_features='auto',  
                                                     max_leaf_nodes=None,  
                                                     min_impurity_decrease=0.0,  
                                                     min_impurity_split=None,  
                                                     min_samples_leaf=1,  
                                                     min_samples_split=2,  
                                                     min_weight_fraction_leaf=0.0,  
                                                     n_estimators='warn', n_jobs=None,  
                                                     oob_score=False, random_state=None,  
                                                     verbose=0, warm_start=False),  
                      iid='warn', n_jobs=-1,  
                      param_grid={'max_depth': [None, 1, 2, 3],  
                                  'n_estimators': [500, 700, 1000]},  
                      pre_dispatch='2*n_jobs', refit=True, return_train_score=False,  
                      scoring=None, verbose=1)
```

```
In [188]: CV_rfr.best_params_
```

```
Out[188]: {'max_depth': None, 'n_estimators': 700}
```

```
In [189]: RFR2 = RandomForestRegressor(random_state=1, max_depth = None, n_estimators = 700)
RFR2.fit(data_X_train, data_y_train)
```

C:\Users\Artem\AppData\Roaming\Python\Python37\site-packages\ipykernel_launcher.py:2: DataConversionWarning: A column -vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

```
Out[189]: RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
                                 max_features='auto', max_leaf_nodes=None,
                                 min_impurity_decrease=0.0, min_impurity_split=None,
                                 min_samples_leaf=1, min_samples_split=2,
                                 min_weight_fraction_leaf=0.0, n_estimators=700,
                                 n_jobs=None, oob_score=False, random_state=1, verbose=0,
                                 warm_start=False)
```

```
In [190]: target_RFR2 = RFR2.predict(data_X_test)
```

```
In [191]: print_stat(data_y_test, target_RFR2)
```

```
#Было:  
#0.004899101427321029  
#0.00027346637497344947  
#0.0009250957384237517
```

```
#Все стало чуть лучше... хотя точность изначально была неплохая: > 99.5% !!!
```

```
0.00460669094962358  
0.0002566624006824704  
0.0008576846533807103
```

AdaBoost

```
In [192]: ABR1 = AdaBoostRegressor(random_state=1, base_estimator = RandomForestRegressor(random_state=1, max_depth = 3, n_estimators = 500))
ABR1.fit(data_X_train, data_y_train)
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
Out[192]: AdaBoostRegressor(base_estimator=RandomForestRegressor(bootstrap=True,
                                                               criterion='mse',
                                                               max_depth=3,
                                                               max_features='auto',
                                                               max_leaf_nodes=None,
                                                               min_impurity_decrease=0.0,
                                                               min_impurity_split=None,
                                                               min_samples_leaf=1,
                                                               min_samples_split=2,
                                                               min_weight_fraction_leaf=0.0,
                                                               n_estimators=500,
                                                               n_jobs=None,
                                                               oob_score=False,
                                                               random_state=1,
                                                               verbose=0,
                                                               warm_start=False),
                             learning_rate=1.0, loss='linear', n_estimators=50,
                             random_state=1)
```

```
In [193]: target_ABR1 = ABR1.predict(data_X_test)
```

```
In [194]: print_stat(data_y_test, target_ABR1)
```

```
0.009126237629093047
0.0003301234655932829
0.006254570383999151
```

```
In [196]: parameters = {'n_estimators': (1, 2, 5), 'base_estimator__max_depth': (1, 2, 3)}
```

```
In [197]: CV_ABR = GridSearchCV(ABR1, parameters)
```

```
In [198]: CV_ABR.fit(data_X_train, data_y_train)
```

```
c:\program files\python37\lib\site-packages\sklearn\model_selection\_split.py:1978: FutureWarning: The default value  
of cv will change from 3 to 5 in version 0.22. Specify it explicitly to silence this warning.  
    warnings.warn(CV_WARNING, FutureWarning)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)  
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y  
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
    y = column_or_1d(y, warn=True)
```



```
y = column_or_1d(y, warn=True)
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
Out[198]: GridSearchCV(cv='warn', error_score='raise-deprecating',
                       estimator=AdaBoostRegressor(base_estimator=RandomForestRegressor(bootstrap=True,
                                                                 criterion='mse',
                                                                 max_depth=3,
                                                                 max_features='auto',
                                                                 max_leaf_nodes=None,
                                                                 min_impurity_decrease=0.0,
                                                                 min_impurity_split=None,
                                                                 min_samples_leaf=1,
                                                                 min_samples_split=2,
                                                                 min_weight_fraction_leaf=0.0,
                                                                 n_estimators=500,
                                                                 n_jobs=None,
                                                                 oob_score=False,
                                                                 random_state=1,
                                                                 verbose=0,
                                                                 warm_start=False),
                       learning_rate=1.0, loss='linear',
                       n_estimators=50, random_state=1),
                       iid='warn', n_jobs=None,
                       param_grid={'base_estimator__max_depth': (1, 2, 3),
                                   'n_estimators': (1, 2, 5)},
                       pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
                       scoring=None, verbose=0)
```

```
In [199]: CV_ABR.best_params_
```

```
Out[199]: {'base_estimator__max_depth': 3, 'n_estimators': 5}
```

```
In [200]: ABR2 = AdaBoostRegressor(random_state=1, base_estimator = RandomForestRegressor(random_state=1, max_depth = 3, n_estimators = 500), n_estimators = 5)
ABR2.fit(data_X_train, data_y_train)
```

```
c:\program files\python37\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
y = column_or_1d(y, warn=True)
```

```
Out[200]: AdaBoostRegressor(base_estimator=RandomForestRegressor(bootstrap=True,
                                                               criterion='mse',
                                                               max_depth=3,
                                                               max_features='auto',
                                                               max_leaf_nodes=None,
                                                               min_impurity_decrease=0.0,
                                                               min_impurity_split=None,
                                                               min_samples_leaf=1,
                                                               min_samples_split=2,
                                                               min_weight_fraction_leaf=0.0,
                                                               n_estimators=500,
                                                               n_jobs=None,
                                                               oob_score=False,
                                                               random_state=1,
                                                               verbose=0,
                                                               warm_start=False),
                            learning_rate=1.0, loss='linear', n_estimators=5,
                            random_state=1)
```

```
In [202]: target_ABR2 = ABR2.predict(data_X_test)
```

```
In [204]: print_stat(data_y_test, target_ABR2)
```

```
#Было  
#0.009126237629093047  
#0.0003301234655932829  
#0.006254570383999151
```

```
#Стало лучше.
```

```
0.007879655388251881  
0.00032933231047914684  
0.003904171936929344
```

Метод группового учета аргументов

```
In [210]: GMDH1 = MultilayerGMDH()
```

```
In [211]: GMDH1.fit(data_X_train, data_y_train)
target_GMDH1 = GMDH1.predict(data_X_test)

c:\program files\python37\lib\site-packages\gmdhpy\data_preprocessing.py:11: FutureWarning: Method .as_matrix will be
removed in a future version. Use .values instead.
    data_x = data_x.as_matrix()
c:\program files\python37\lib\site-packages\gmdhpy\data_preprocessing.py:14: FutureWarning: Method .as_matrix will be
removed in a future version. Use .values instead.
    data_y = data_y.as_matrix()

train layer0 in 1.95 sec
train layer1 in 7.70 sec
train layer2 in 7.81 sec
train layer3 in 7.78 sec
train layer4 in 7.68 sec
train layer5 in 7.71 sec
train layer6 in 7.94 sec
train layer7 in 9.92 sec
train layer8 in 8.18 sec
train layer9 in 8.21 sec
train layer10 in 8.30 sec
train layer11 in 8.24 sec

c:\program files\python37\lib\site-packages\gmdhpy\data_preprocessing.py:60: FutureWarning: Method .as_matrix will be
removed in a future version. Use .values instead.
    data_x = data_x.as_matrix()
```

```
In [212]: print_stat(data_y_test, target_GMDH_1)
```

```
0.005862110066783228
0.00025267746965485665
0.0023521028893151792
```

```
In [213]: GMDH2 = MultilayerGMDH(ref_functions=['linear_cov', 'quadratic', 'cubic'], seq_type='random')
GMDH2.fit(data_X_train, data_y_train)
target_GMDH2 = GMDH2.predict(data_X_test)

c:\program files\python37\lib\site-packages\gmdhpy\data_preprocessing.py:11: FutureWarning: Method .as_matrix will be
removed in a future version. Use .values instead.
    data_x = data_x.as_matrix()
c:\program files\python37\lib\site-packages\gmdhpy\data_preprocessing.py:14: FutureWarning: Method .as_matrix will be
removed in a future version. Use .values instead.
    data_y = data_y.as_matrix()

train layer0 in 11.06 sec
train layer1 in 36.68 sec
train layer2 in 35.04 sec
train layer3 in 39.28 sec
train layer4 in 40.59 sec
train layer5 in 33.92 sec
train layer6 in 34.16 sec

c:\program files\python37\lib\site-packages\gmdhpy\data_preprocessing.py:60: FutureWarning: Method .as_matrix will be
removed in a future version. Use .values instead.
    data_x = data_x.as_matrix()
```

```
In [215]: print_stat(data_y_test, target_GMDH2)
```

```
#Было
#0.005862110066783228
#0.00025267746965485665
#0.0023521028893151792
```

```
#Стало
```

```
0.006731036287392927
0.0002562126497196879
0.0034648578009456397
```

Модель показывает хорошие метрики, но до подбора гиперпараметров они были лучше

Анализ

1 - KNN

2 - Линейная

3 - Опорные векторы

4 - Случайный лес

5 - AdaBoost

6 - Метод группового учета аргументов (GMDH)

```
In [224]: d2 = [{"model_№": 1, "model": "KNN", "mean_absolute_error" : mean_absolute_error(data_y_test, target_KNN2), "median_absolute_error": median_absolute_error(data_y_test, target_KNN2),
             "mean_squared_error": mean_squared_error(data_y_test, target_KNN2)}, {"model_№": 2, "model": "LR", "mean_absolute_error" : mean_absolute_error(data_y_test, target_LR2), "median_absolute_error": median_absolute_error(data_y_test, target_LR2),
             "mean_squared_error": mean_squared_error(data_y_test, target_LR2)}, {"model_№": 3, "model": "SVR", "mean_absolute_error" : mean_absolute_error(data_y_test, target_SVR2), "median_absolute_error": median_absolute_error(data_y_test, target_SVR2),
             "mean_squared_error": mean_squared_error(data_y_test, target_SVR2)}, {"model_№": 4, "model": "RFR", "mean_absolute_error" : mean_absolute_error(data_y_test, target_RFR2), "median_absolute_error": median_absolute_error(data_y_test, target_RFR2),
             "mean_squared_error": mean_squared_error(data_y_test, target_RFR2)}, {"model_№": 5, "model": "AB", "mean_absolute_error" : mean_absolute_error(data_y_test, target_ABR2), "median_absolute_error": median_absolute_error(data_y_test, target_ABR2),
             "mean_squared_error": mean_squared_error(data_y_test, target_ABR2)}, {"GS": 'после', "model_№": 6, "model": "GMDH", "mean_absolute_error" : mean_absolute_error(data_y_test, target_GMDH2), "median_absolute_error": median_absolute_error(data_y_test, target_GMDH2),
             "mean_squared_error": mean_squared_error(data_y_test, target_GMDH2)}]
```

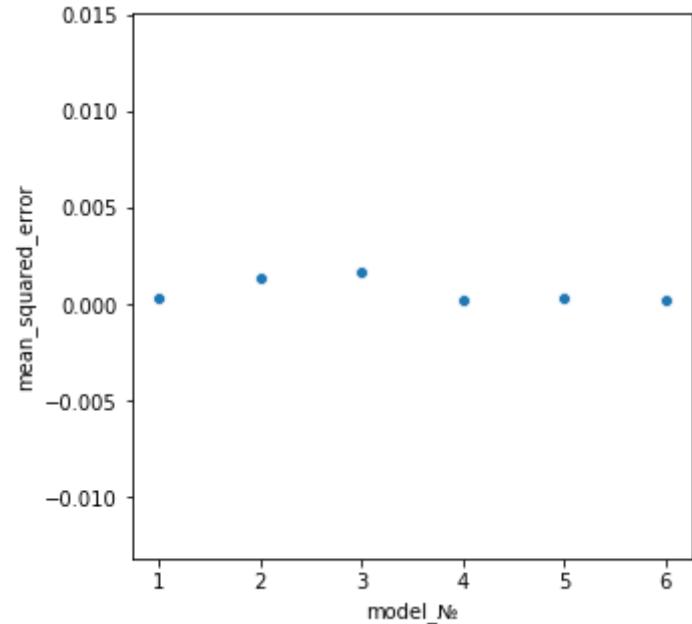
```
In [225]: dd2 = pd.DataFrame(d2)
```

```
In [226]: print(dd2)
```

	GS	mean_absolute_error	mean_squared_error	median_absolute_error	\
0	NaN	0.005551	0.000313	0.001104	
1	NaN	0.019139	0.001377	0.013695	
2	NaN	0.030476	0.001656	0.025142	
3	NaN	0.004607	0.000257	0.000858	
4	NaN	0.007880	0.000329	0.003904	
5	после	0.006731	0.000256	0.003465	
	model	model_№			
0	KNN	1			
1	LR	2			
2	SVR	3			
3	RFR	4			
4	AB	5			
5	GMDH	6			

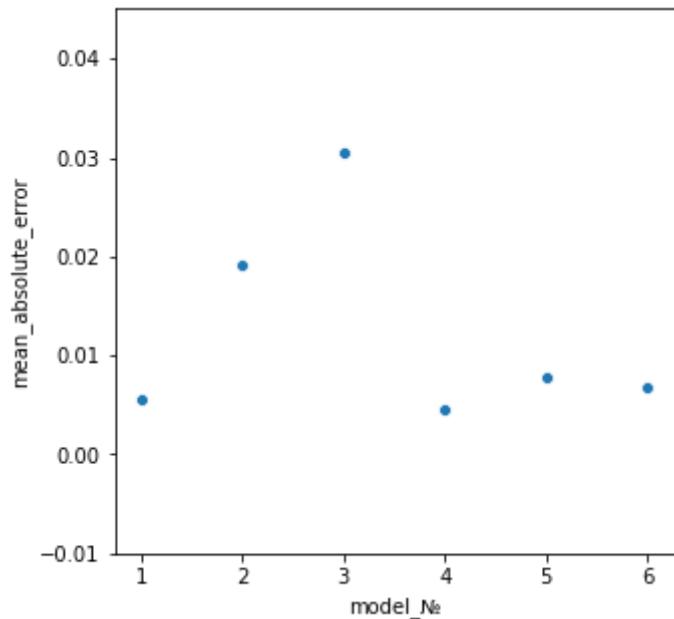
```
In [227]: fig, ax = plt.subplots(figsize=(5,5))
sns.scatterplot(ax=ax, x='model_Nº', y='mean_squared_error', data=dd2)
```

```
Out[227]: <matplotlib.axes._subplots.AxesSubplot at 0x24bf24b00b8>
```



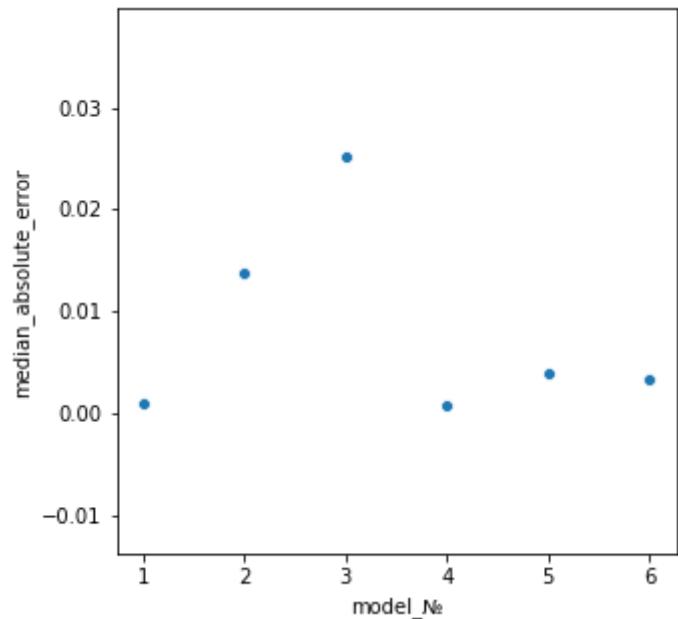
```
In [228]: fig, ax = plt.subplots(figsize=(5,5))
sns.scatterplot(ax=ax, x='model_Nº', y='mean_absolute_error', data=dd2)
```

```
Out[228]: <matplotlib.axes._subplots.AxesSubplot at 0x24bf24b06a0>
```



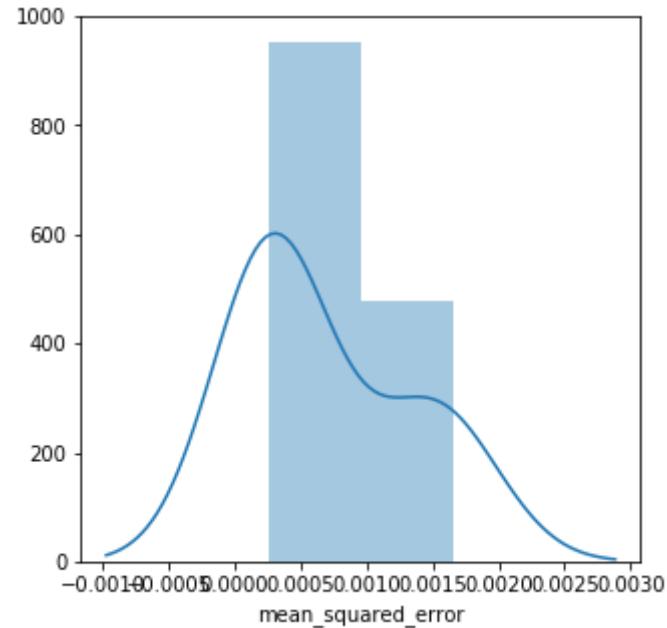
```
In [229]: fig, ax = plt.subplots(figsize=(5,5))
sns.scatterplot(ax=ax, x='model_Nº', y='median_absolute_error', data=dd2)
```

```
Out[229]: <matplotlib.axes._subplots.AxesSubplot at 0x24bf4f20048>
```



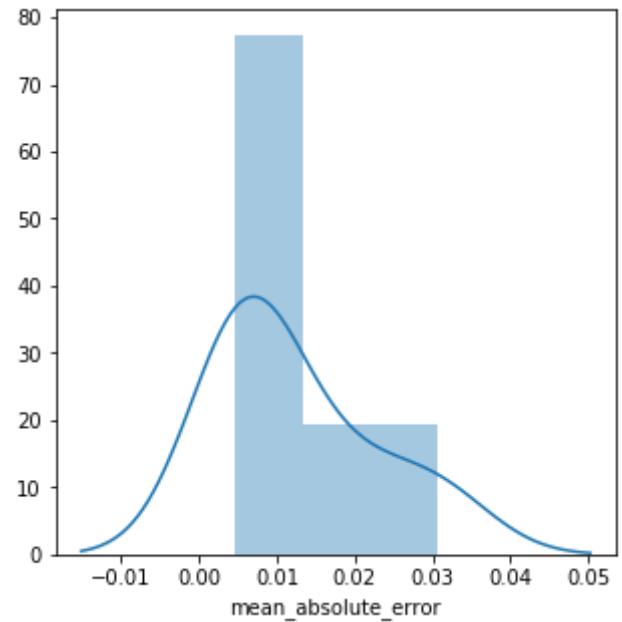
```
In [230]: #Гистограмма Позволяет оценить плотность вероятности распределения данных  
fig, ax = plt.subplots(figsize=(5,5))  
sns.distplot(dd2['mean_squared_error'])
```

```
Out[230]: <matplotlib.axes._subplots.AxesSubplot at 0x24bf4cfbd68>
```



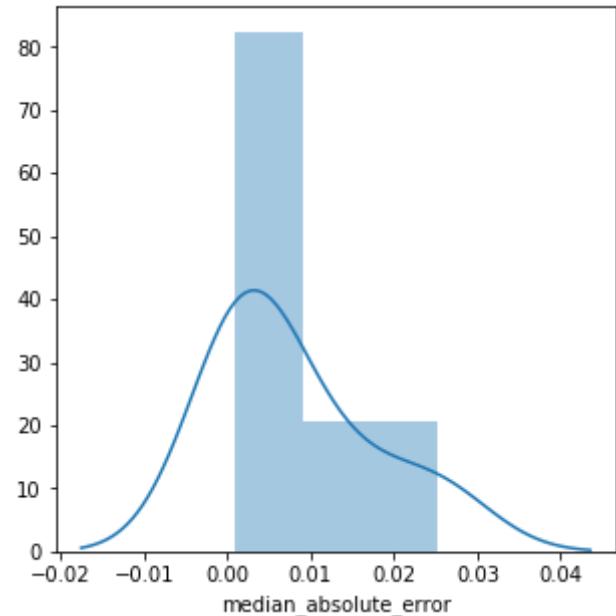
```
In [231]: fig, ax = plt.subplots(figsize=(5,5))
sns.distplot(dd2['mean_absolute_error'])
```

```
Out[231]: <matplotlib.axes._subplots.AxesSubplot at 0x24bf4f97390>
```



```
In [232]: fig, ax = plt.subplots(figsize=(5,5))
sns.distplot(dd2['median_absolute_error'])
```

```
Out[232]: <matplotlib.axes._subplots.AxesSubplot at 0x24ba29d9048>
```

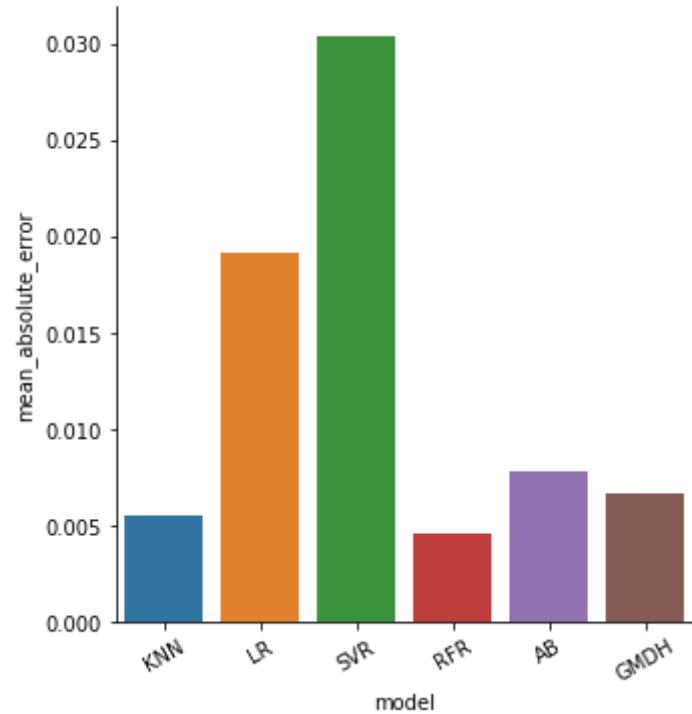


```
In [233]: g = sns.factorplot(x='model'
                           ,y= 'mean_absolute_error'
                           ,data=dd2
                           ,kind='bar'

                           )
g.set_xticklabels(rotation=30)
```

```
c:\program files\python37\lib\site-packages\seaborn\categorical.py:3666: UserWarning: The `factorplot` function has been renamed to `catplot`. The original name will be removed in a future release. Please update your code. Note that the default `kind` in `factorplot` (`'point'`) has changed `'strip'` in `catplot`.
warnings.warn(msg)
```

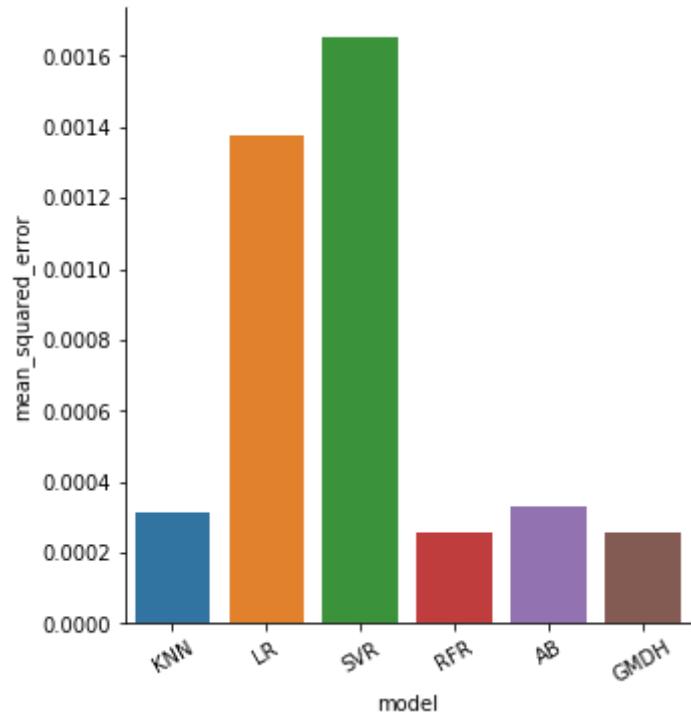
```
Out[233]: <seaborn.axisgrid.FacetGrid at 0x24bf4d81748>
```



```
In [234]: g = sns.factorplot(x='model'
                           ,y= 'mean_squared_error'
                           ,data=dd2
                           ,kind='bar'

                           )
g.set_xticklabels(rotation=30)
```

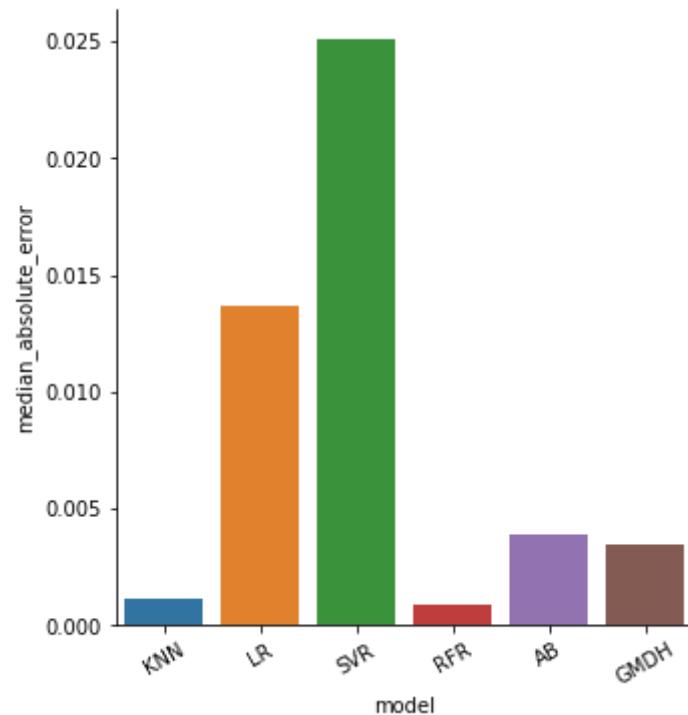
Out[234]: <seaborn.axisgrid.FacetGrid at 0x24bf4f857f0>



```
In [235]: g = sns.factorplot(x='model'
                           ,y= 'median_absolute_error'
                           ,data=dd2
                           ,kind='bar'

                           )
g.set_xticklabels(rotation=30)
```

```
Out[235]: <seaborn.axisgrid.FacetGrid at 0x24bf3693668>
```



Вывод: при анализе метрик почти всегда лидирует RandomForestRegressor модель. За ней идут метод К ближайших соседей. В данном случае лучше всего использовать RandomForestRegressor модель. Так же можно заметить, что подбор гиперпараметров почти всегда улучшает метрики.