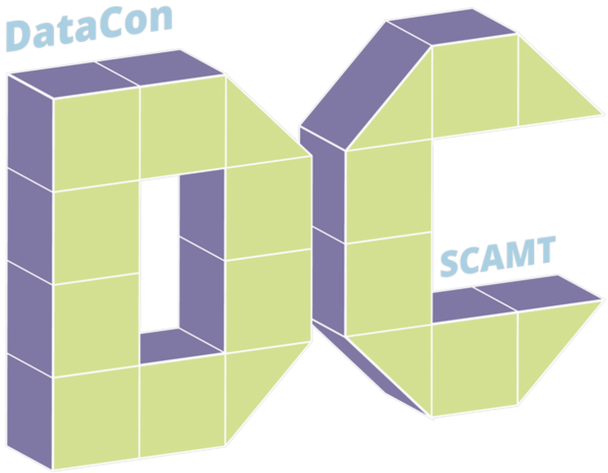
GEEK BATTLE

Предсказание токсичности наноматериалов.

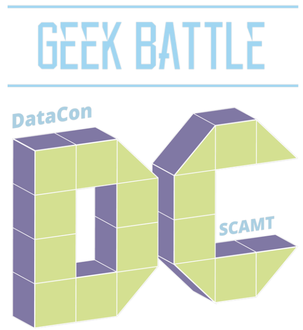
Создание модели для предсказания свойств наночастиц: предсказание cell viability.



Евгения Полежаева,
Ксения Парутина,
Валерий Древлянский.

План работы

1. Установка библиотек
2. Импорт данных
3. Предобработка данных
4. Создание единой базы данных
5. Анализ
6. Модели
7. Вывод



1. Установка библиотек

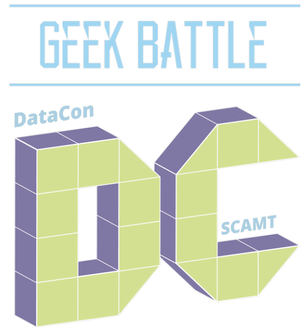
Тут все понятно и очевидно..

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


from sklearn.model_selection import GridSearchCV
#from catboost import CatBoostClassifier
from sklearn.model_selection import train_test_split

import sklearn
from sklearn.ensemble import RandomForestRegressor
from sklearn import preprocessing
```

2. Импорт данных

Что мы имеем: 5 таблиц с **разным** набором признаков.

Необходимо переименовать признаки в каждом из DataFrame's и объединить их (получение уникальных значений и создание единой БД).



3.1. Предобработка данных

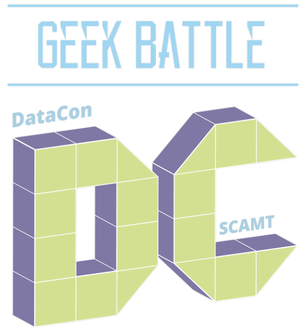
Database_1 :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 494 entries, 0 to 493
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   material            494 non-null    object
 1   elements            494 non-null    object
 2   electronegativity   493 non-null    float64
 3   ionic_radius        493 non-null    float64
 4   core_size           494 non-null    float64
 5   size_in_water       493 non-null    float64
 6   surface_charge      494 non-null    float64
 7   surface_area        493 non-null    float64
 8   cell_type           494 non-null    int64
 9   concentration       494 non-null    float64
 10  number_of_atoms     494 non-null    int64
 11  mw                  493 non-null    float64
 12  tps                 494 non-null    float64
 13  a                   493 non-null    float64
 14  b                   494 non-null    float64
 15  c                   493 non-null    float64
 16  alpha               494 non-null    int64
 17  beta                494 non-null    float64
 18  gama                494 non-null    int64
 19  density             494 non-null    float64
 20  viability           494 non-null    float64
dtypes: float64(15), int64(4), object(2)
memory usage: 81.2+ KB
None
['CuO' 'ZnO' 'Mn2O3' 'CoO' 'CeO2' 'Fe2O3' 'Gd2O3' 'HfO2' 'In2O3' 'La2O3'
 'NiO' 'Sb2O3' 'SiO2' 'Al2O3']
['Cu' 'Zn' 'Mn' 'Co' 'cobalt' 'Ce' 'Fe' 'Gd' 'Hf' 'In' 'La' 'Ni' 'Sb' 'Si'
 'Al' 'Iron']
```

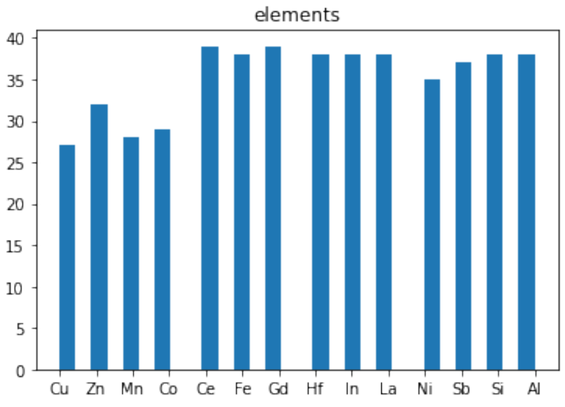
В 1 таблице (*Database_1*) все материалы-***неорганические***.

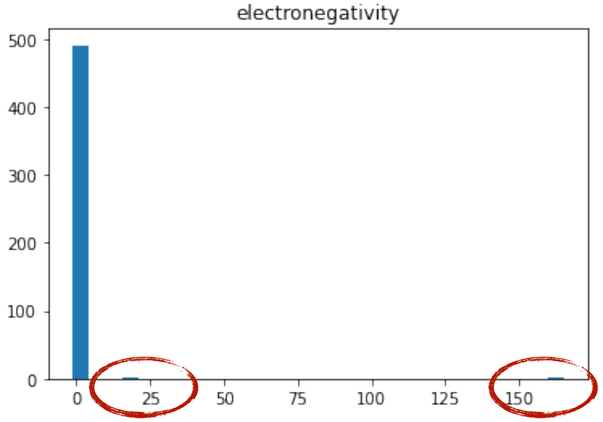
Создадим соответствующий признак.

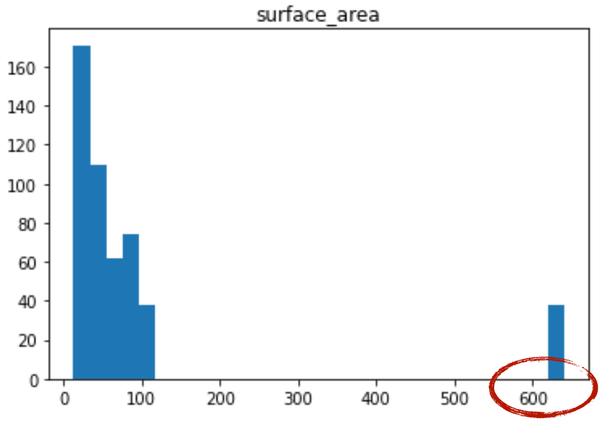
Большинство признаков - числовые. Необходимо посмотреть их распределение.

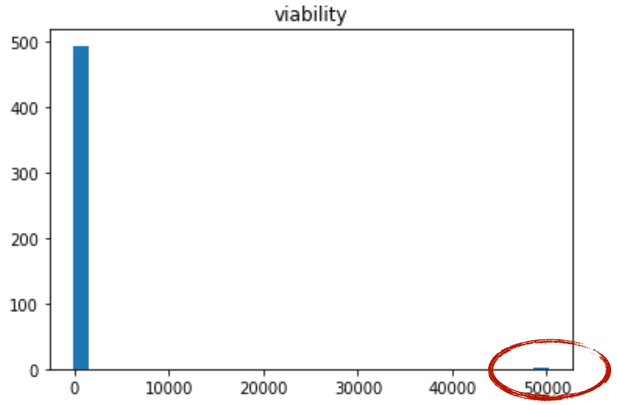


3.1. Предобработка данных



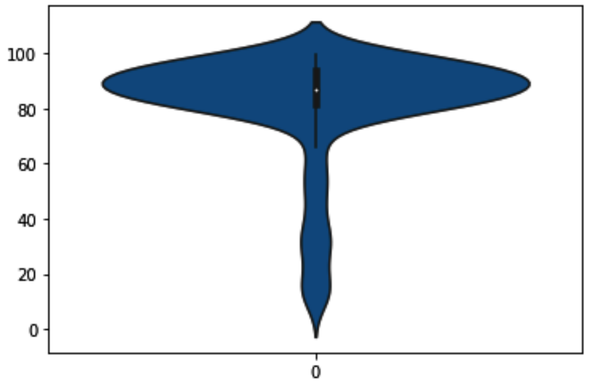




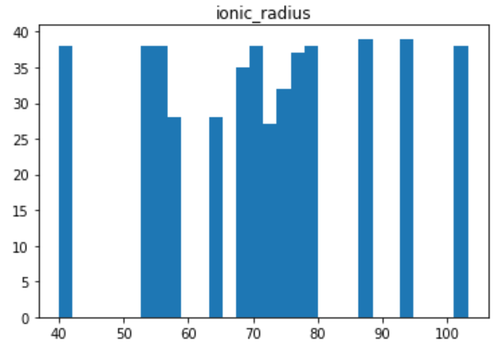


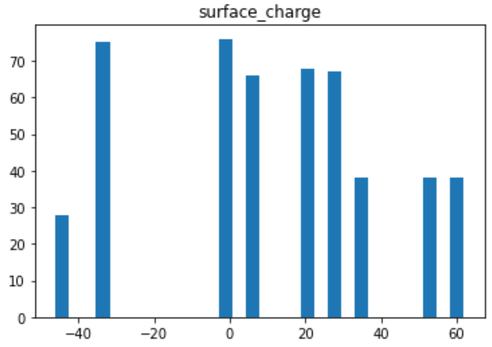

```
#Viability > 0
db_1.viability = db_1.viability[db_1['viability']<10000]
db_1.viability = db_1.viability.abs()
sns.violinplot(data=db_1.viability)
```

```
<AxesSubplot:>
```



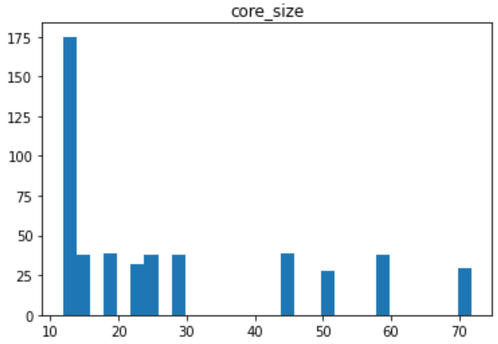
3.1. Предобработка данных

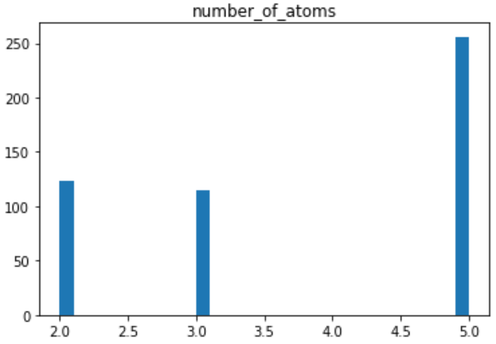


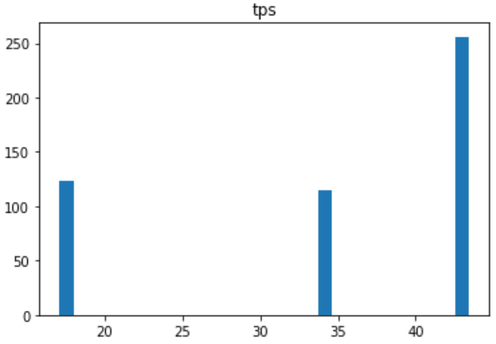


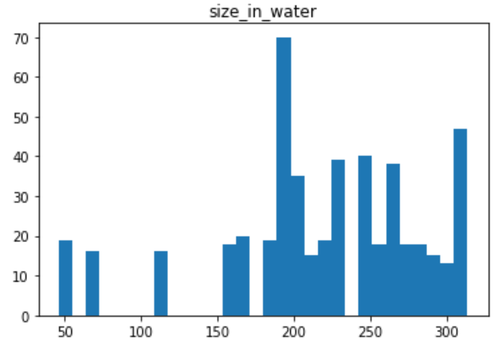
Database_1

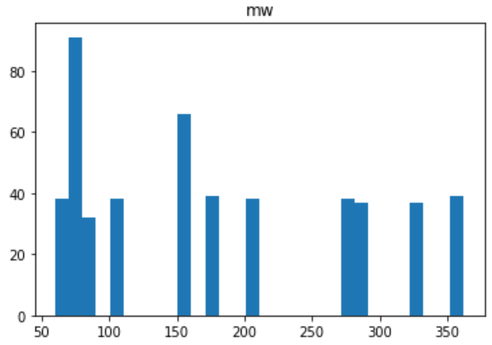
- **Ionic radius, Core_size(nm), size_in_water, Surface_charge, Exposure_dose, Number of atoms, Mw, Tps, density**, - без выбросов.

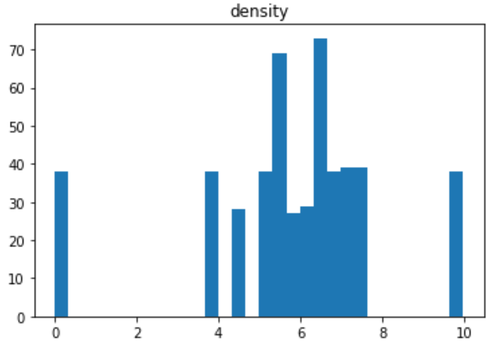












GEEK BATTLE
DataCon
SCAMT

3.2. Предобработка данных

Database_2 :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2896 entries, 0 to 2895
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   material              2896 non-null   object
 1   type_inorganic        2896 non-null   object
 2   coat                  1052 non-null   object
 3   diameter              2896 non-null   float64
 4   concentration         2896 non-null   float64
 5   surface_charge        1261 non-null   float64
 6   cell_type             2896 non-null   object
 7   cell_line             2896 non-null   object
 8   human                 2896 non-null   object
 9   animal                651 non-null    object
 10  cell_morphology       2895 non-null   object
 11  cell_age              2895 non-null   object
 12  source                2896 non-null   object
 13  exposure_time         2896 non-null   int64
 14  test                  2895 non-null   object
 15  test_indicator        2895 non-null   object
 16  biochemical_metric    2895 non-null   object
 17  viability             2896 non-null   float64
 18  interference          2896 non-null   object
 19  colloidal_stability   2896 non-null   object
 20  positive_control      2896 non-null   object
dtypes: float64(4), int64(1), object(16)
memory usage: 475.2+ KB
```

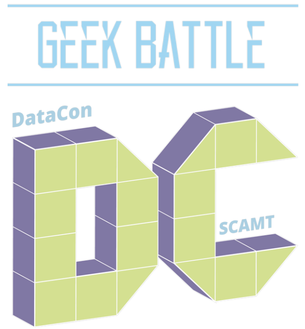
В 2 таблице (*Database_2*) были произведены следующие изменения:

1. Удаление непонятных оксидов (Copper Oxide, Zinc oxide)

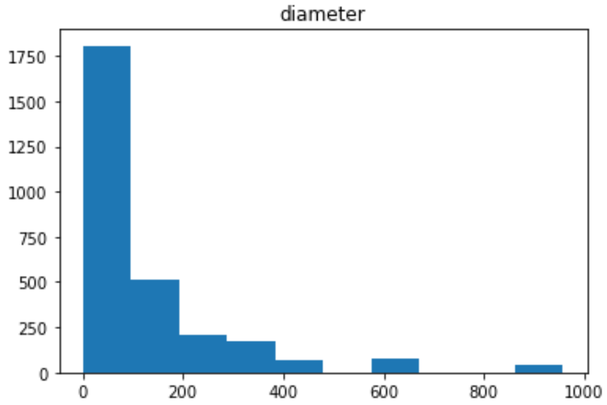
2. Замена «Iron oxide» на «Fe3O4» (Проведен поиск: статья - ссылка на реактив - сайт производителя реактива).

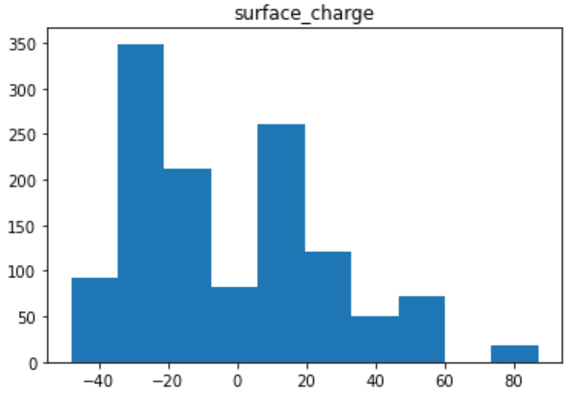
3. Проверка уникальных значений нечисловых признаков и замена на «0» и «1», где это было допустимо.

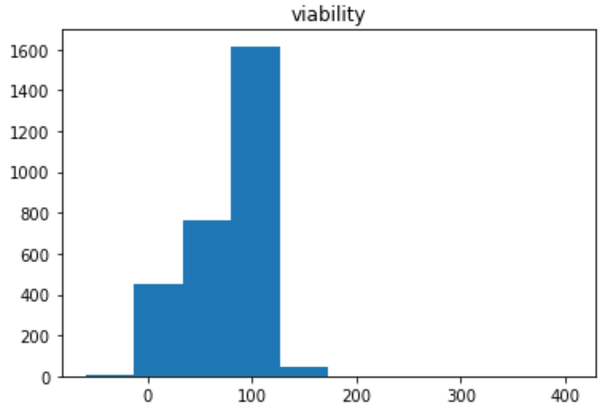
4. Анализ распределения числовых признаков.

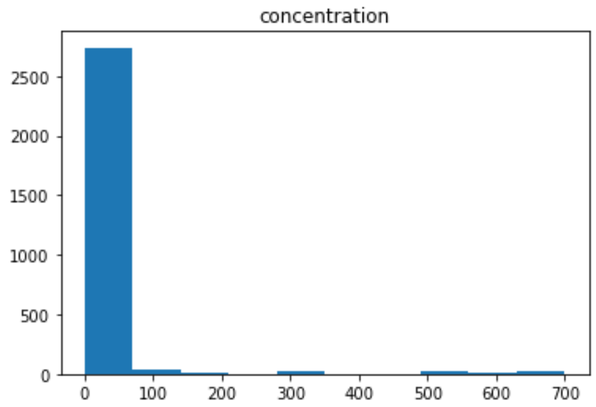


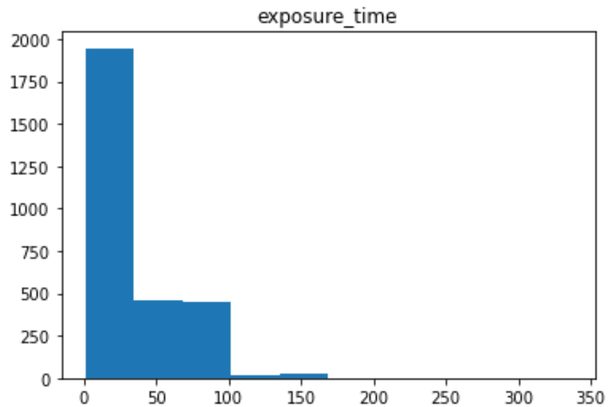
3.2. Предобработка данных





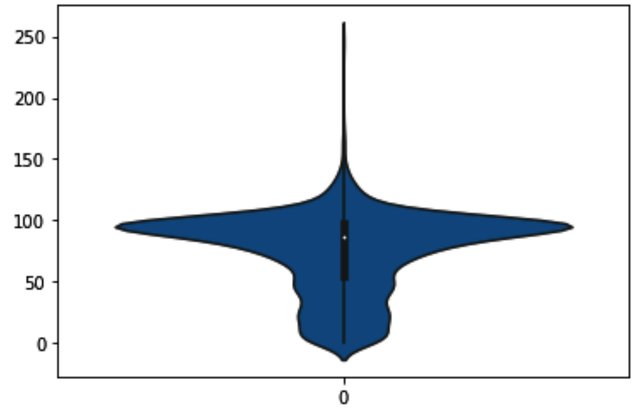







```
db_2.viability = db_2['viability'][db_2['viability']<300]
db_2.viability = db_2.viability.abs()

sns.violinplot(data=db_2.viability)
```

```
<AxesSubplot:>
```



3.3. Предобработка данных

Database_3 :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4111 entries, 0 to 4110
Data columns (total 25 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   material          4111 non-null   object
 1   type_inorganic    4111 non-null   object
 2   shape             4111 non-null   object
 3   coat              4110 non-null   object
 4   synthesis_method  4111 non-null   object
 5   surfacecharge     3112 non-null   object
 6   diameter          3278 non-null   float64
 7   size_in_water     2330 non-null   float64
 8   size_in_medium    1989 non-null   float64
 9   surface_charge    2487 non-null   object
 10  zeta_in_medium    1670 non-null   float64
 11  cell_type         4111 non-null   object
 12  number_of_cells   3684 non-null   float64
 13  human             4111 non-null   object
 14  animal            4110 non-null   object
 15  source            4111 non-null   object
 16  cell_morphology   4111 non-null   object
 17  cell_age          4111 non-null   object
 18  cell_line         4111 non-null   object
 19  exposure_time     4111 non-null   int64
 20  concentration     3889 non-null   float64
 21  test              4111 non-null   object
 22  test_indicator    4111 non-null   object
 23  aspect_ratio      380 non-null    float64
 24  viability         4111 non-null   float64
dtypes: float64(8), int64(1), object(16)
memory usage: 803.1+ KB
```

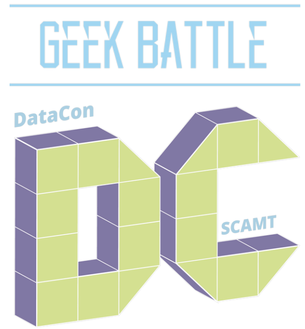
В 3 таблице (*Database_3*) были произведены следующие изменения:

1. Признак 'aspect ratio', 'zeta_in_medium' был удален, т.к. значений очень немного.

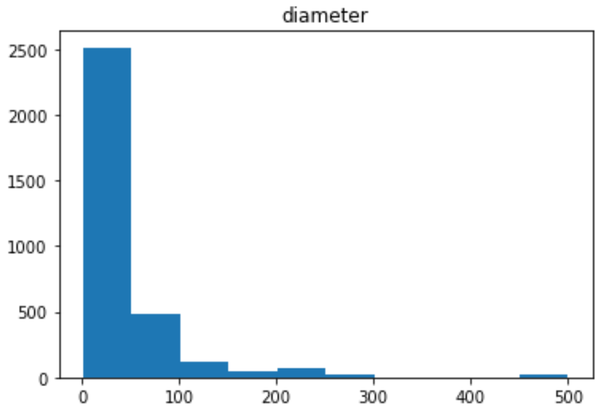
2. Замена «Iron oxide» на «Fe3O4» (Проведен поиск: статья - ссылка на реактив - сайт производителя реактива).

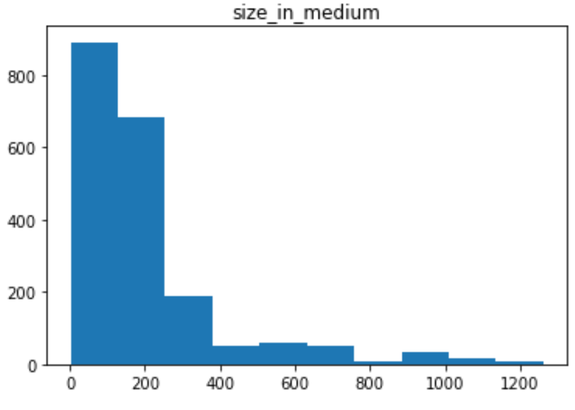
3. Проверка уникальных значений нечисловых признаков и замена на «0» и «1», где это было допустимо.

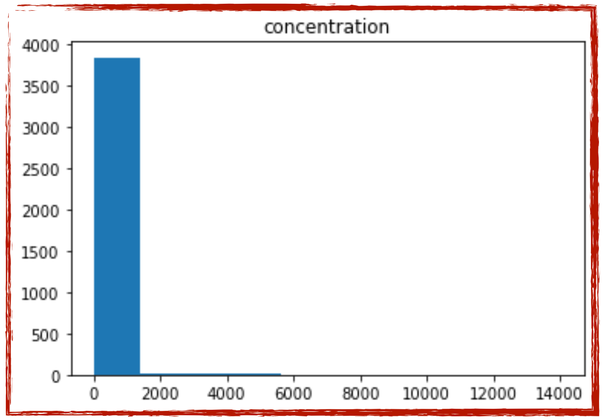
4. Анализ распределения числовых признаков.

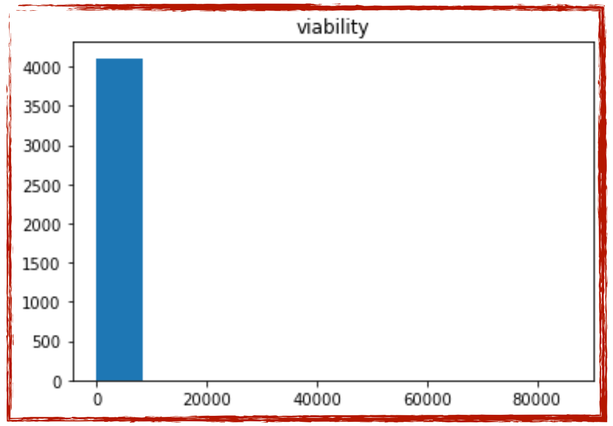


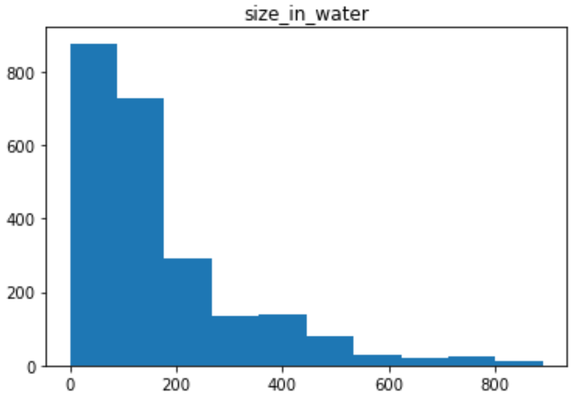
3.3. Предобработка данных

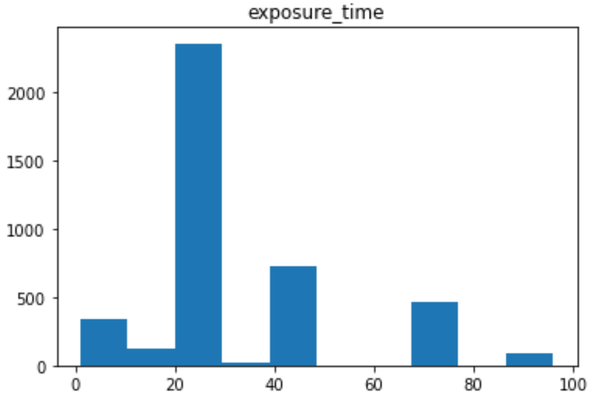


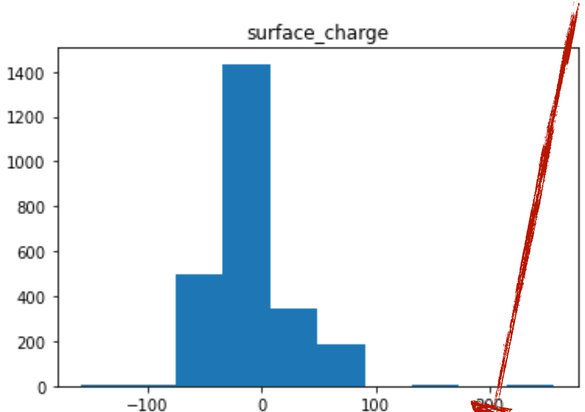












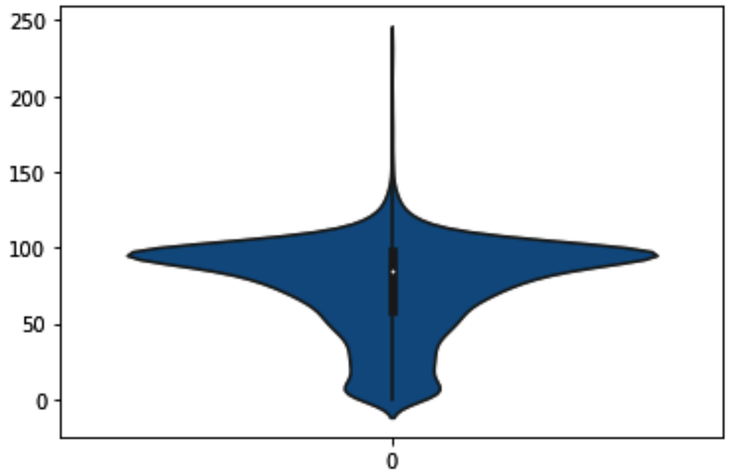
Database_3

Подозрительные значения встречаются в признаке *concentration* и *viability* .

3.3. Предобработка данных

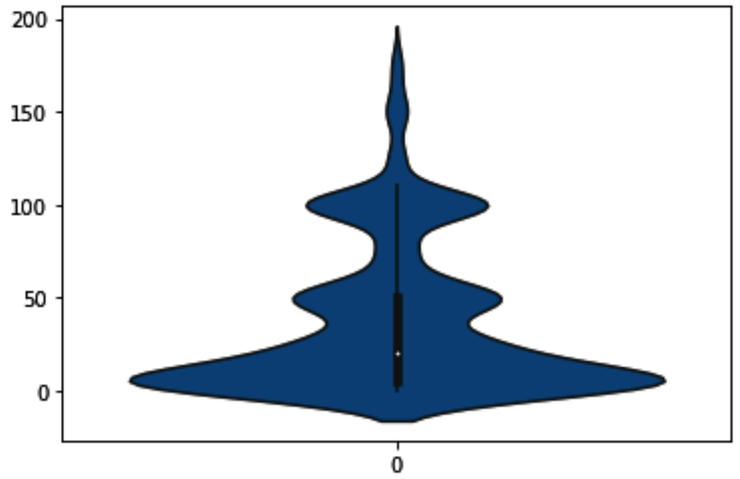
```
db_3.viability = db_3['viability'][db_3['viability']<1000]
db_3.viability = db_3.viability.abs()
sns.violinplot(data=db_3.viability)
```

```
<AxesSubplot:>
```




```
db_3.concentration = db_3['concentration'][db_3['concentration']<200]
sns.violinplot(data=db_3.concentration)
```

```
<AxesSubplot:>
```



Подозрительные значения встречаются в признаке *concentration* и *viability* .

3.4. Предобработка данных

Database_4 :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1068 entries, 0 to 1067
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   material            1066 non-null   object 
 1   elements            1068 non-null   object 
 2   electronegativity   1068 non-null   float64
 3   ionic_radius        1068 non-null   float64
 4   core_size           1066 non-null   float64
 5   size_in_water       1066 non-null   float64
 6   surface_charge      1066 non-null   float64
 7   surface_area        1067 non-null   float64
 8   cell_type           1068 non-null   int64  
 9   concentration       1067 non-null   float64
 10  number_of_atoms     1065 non-null   float64
 11  mw                  1065 non-null   float64
 12  tps                 491 non-null    float64
 13  a                   491 non-null    object 
 14  b                   491 non-null    object 
 15  c                   491 non-null    object 
 16  alpha               491 non-null    object 
 17  beta                491 non-null    object 
 18  gama                491 non-null    object 
 19  density             491 non-null    object 
 20  viability           1068 non-null   float64
dtypes: float64(11), int64(1), object(9)
memory usage: 175.3+ KB
```

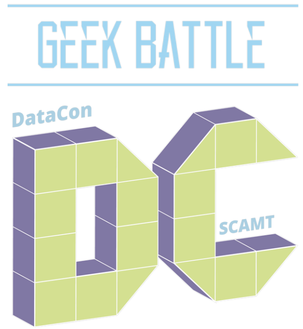
В 4 таблице (*Database_4*) были произведены следующие изменения:

1. Очистка ошибок: удаление неизвестного материала.

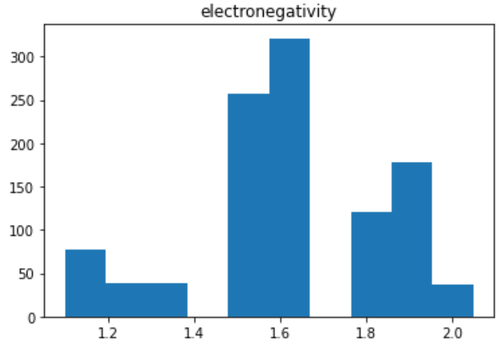
2. Создание признака.

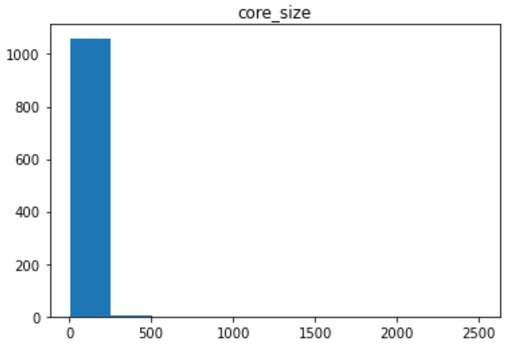
3. Замена "-" на Nan (не полностью правомерно)

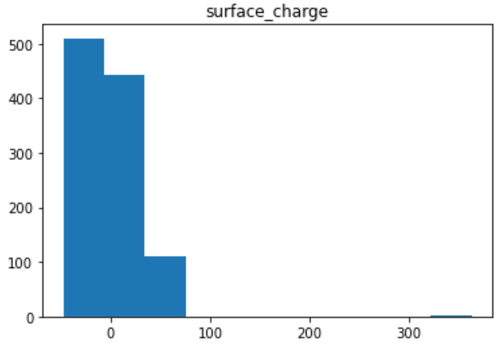
4. Анализ распределения числовых признаков.

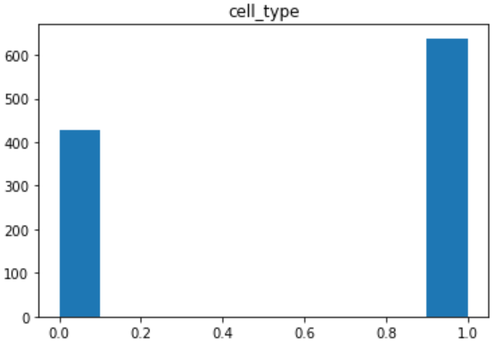


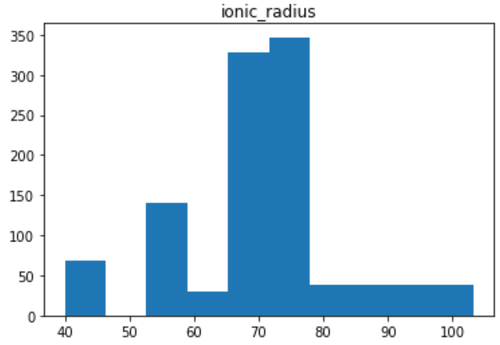
3.4. Предобработка данных

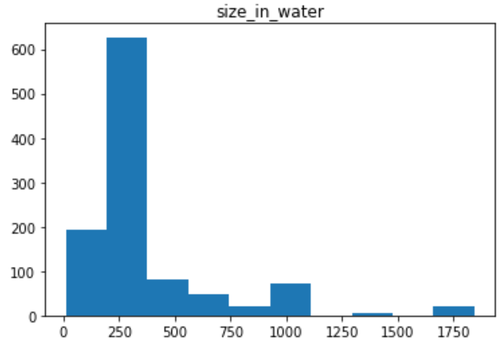


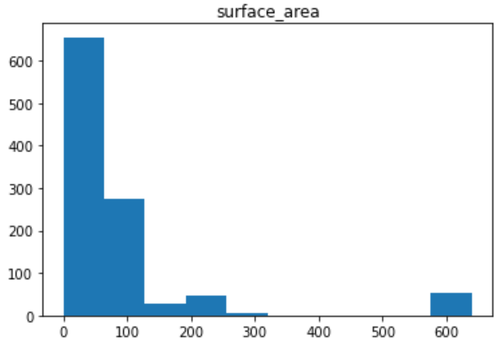


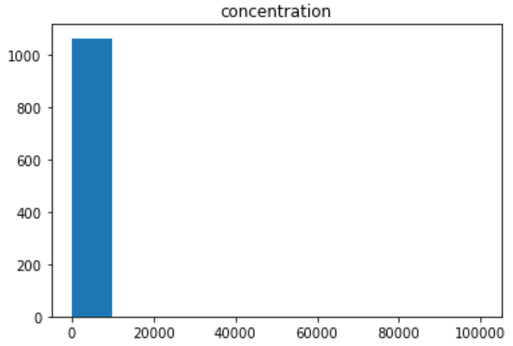


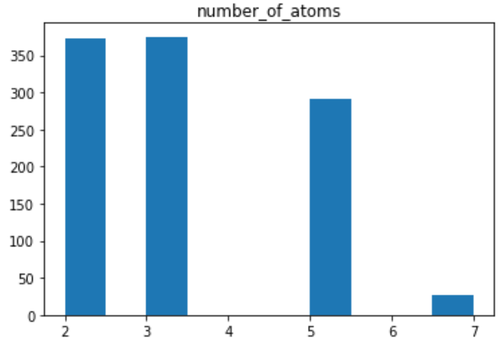


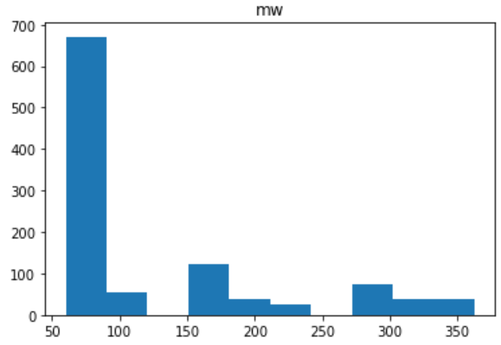


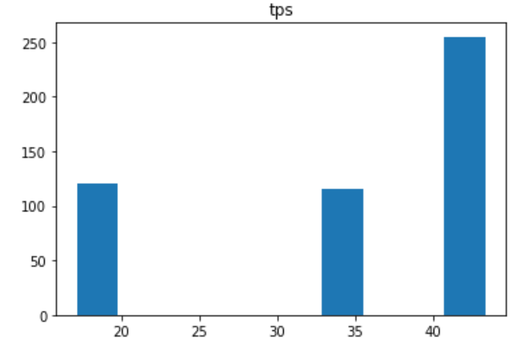


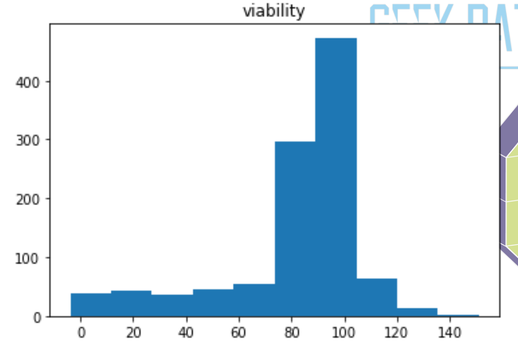












3.5. Предобработка данных

Database_5 :

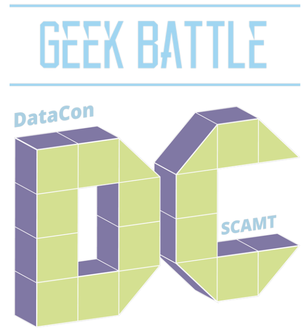
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 574 entries, 0 to 573
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   material         574 non-null    object
 1   core_size        572 non-null    float64
 2   size_in_water    574 non-null    float64
 3   surface_charge   573 non-null    float64
 4   surface_area     574 non-null    float64
 5   cell_line        572 non-null    object
 6   animal           574 non-null    object
 7   source           574 non-null    object
 8   cell_type        574 non-null    object
 9   exposure_time    574 non-null    int64
 10  concentration    574 non-null    float64
 11  viability        574 non-null    float64
 12  toxicity         574 non-null    object
dtypes: float64(6), int64(1), object(6)
memory usage: 58.4+ KB
```

В 5 таблице (*Database_5*) были произведены следующие изменения:

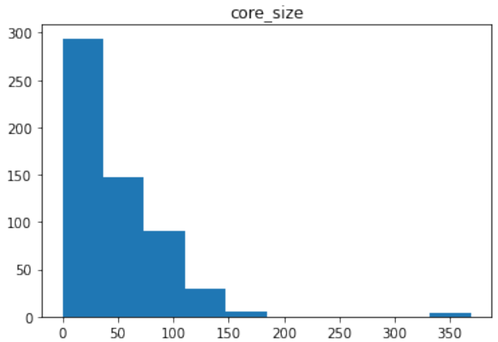
1. Очистка ошибок

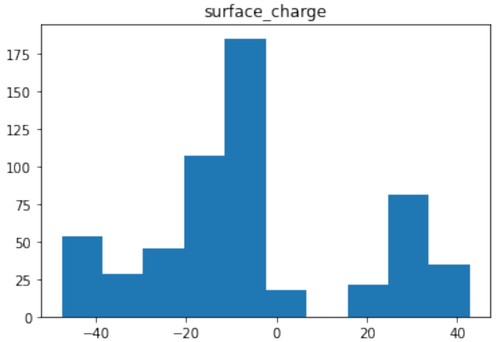
2. Создание признака.

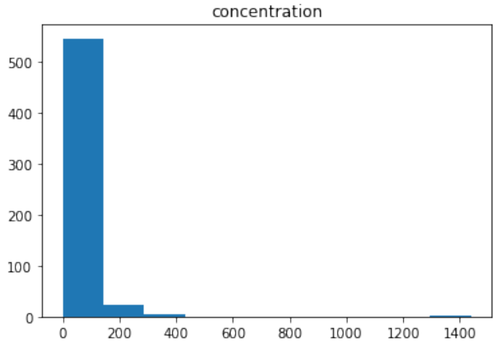
3. Анализ распределения числовых признаков.

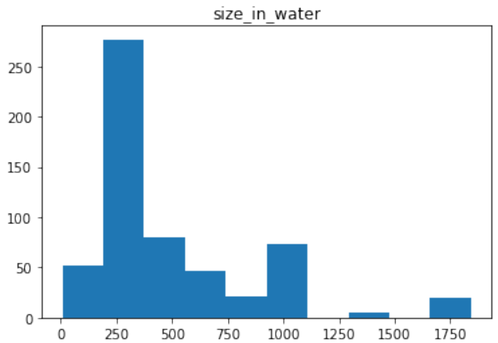


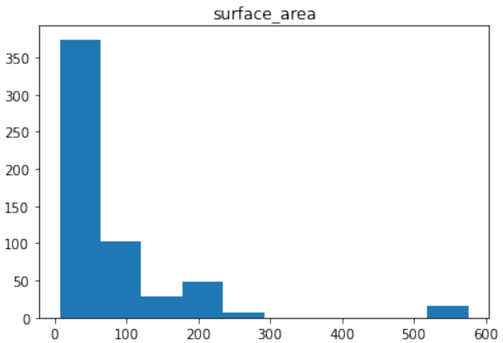
3.5. Предобработка данных

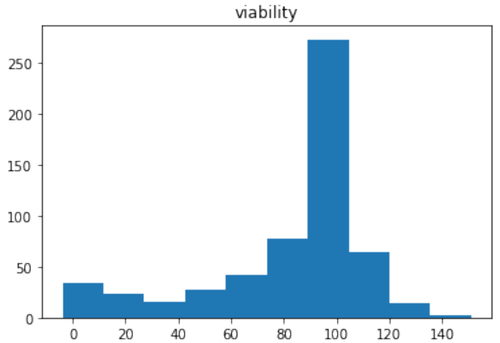


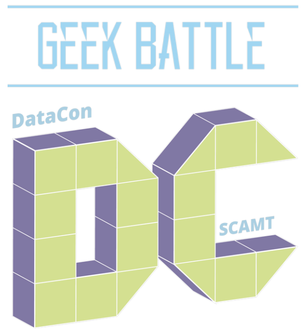




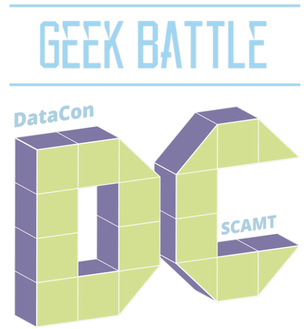






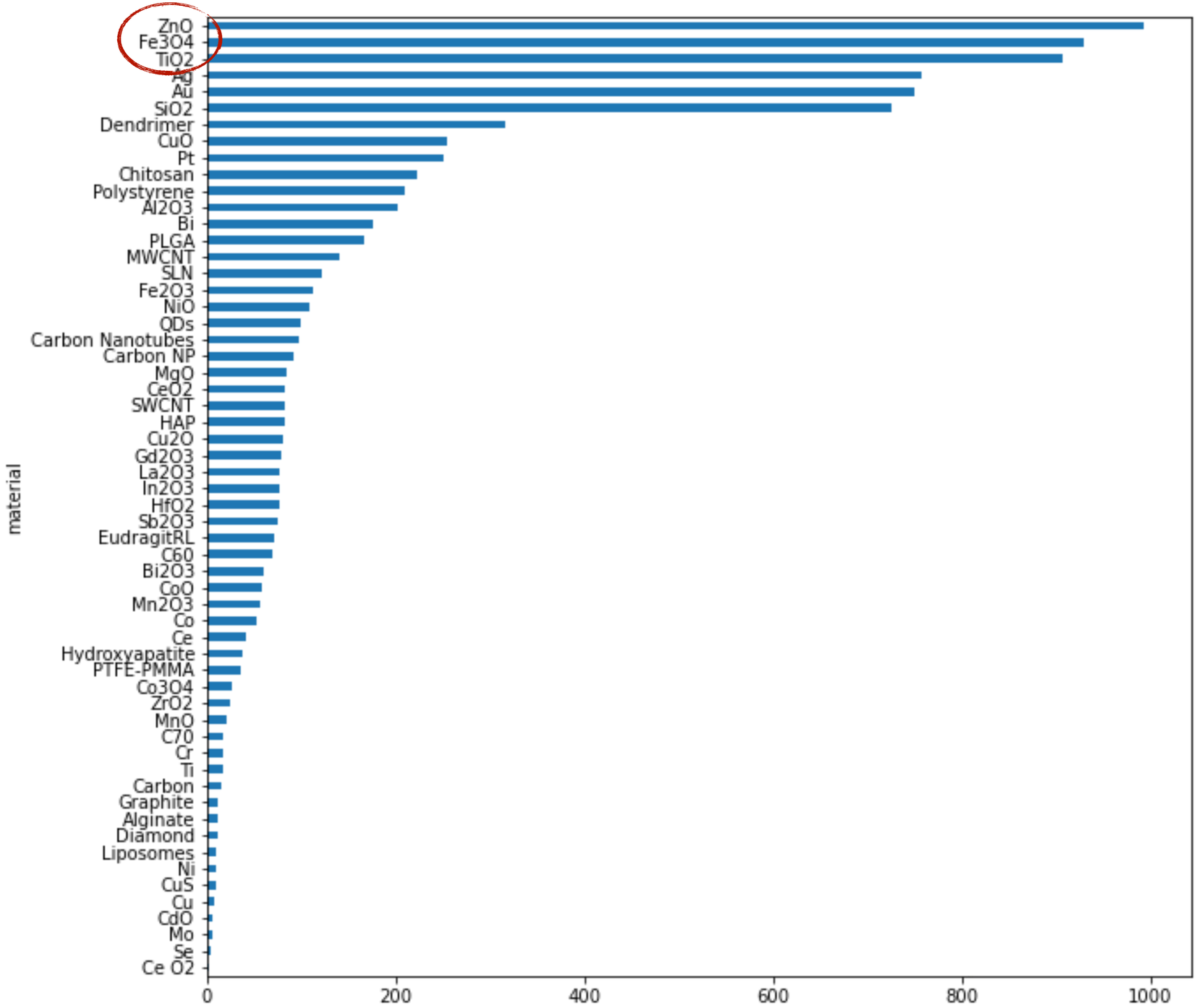


4. Создание единой базы данных - объединение таблиц.



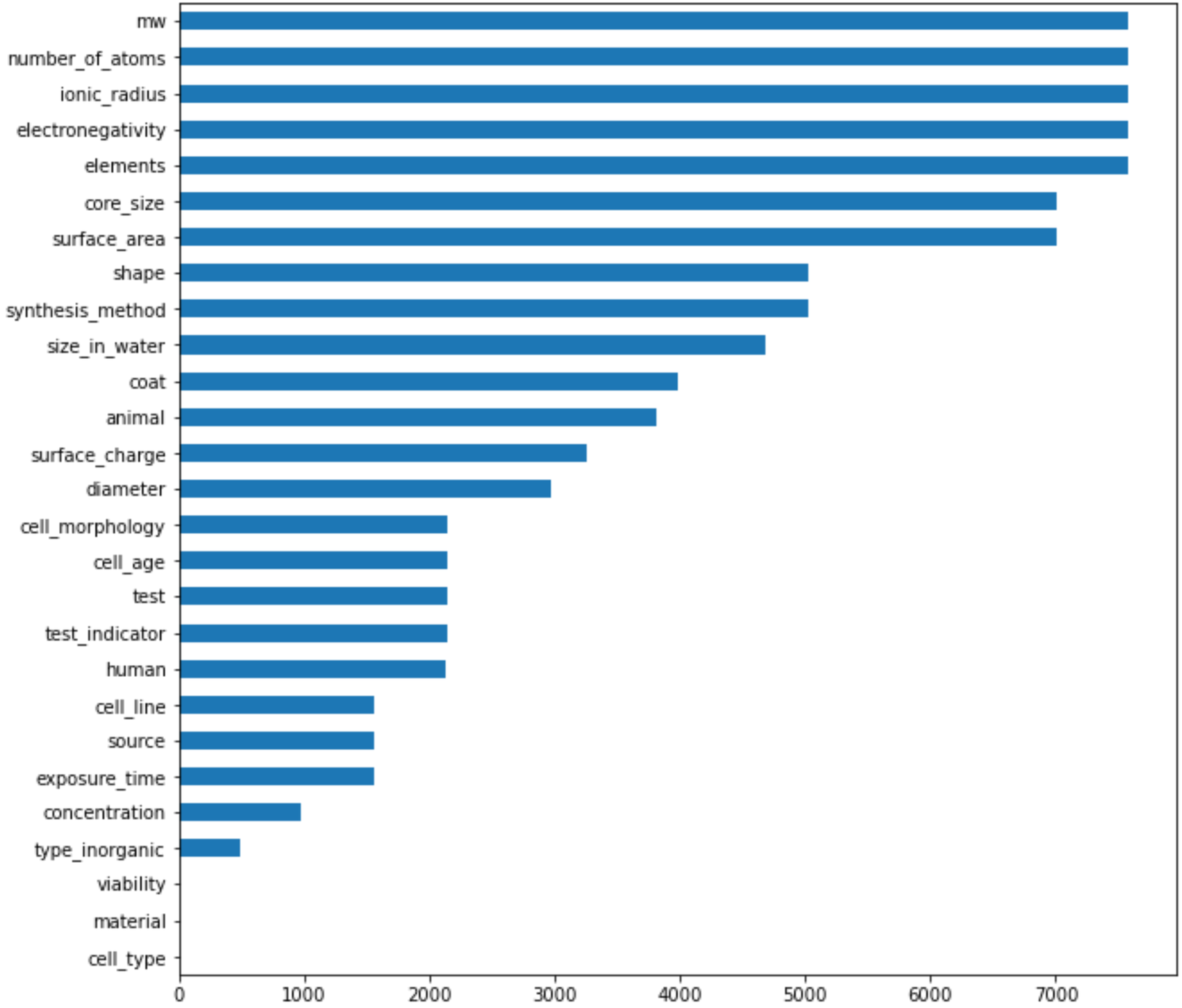
5. Анализ

- **Распределение данных по типу материала:**



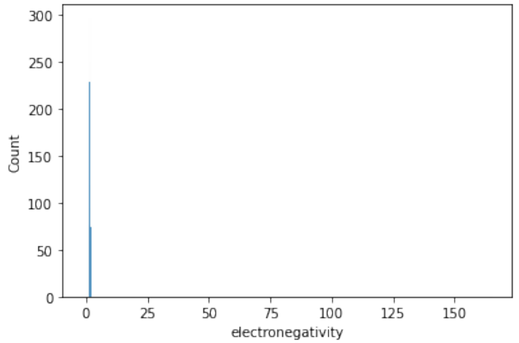
5. Анализ

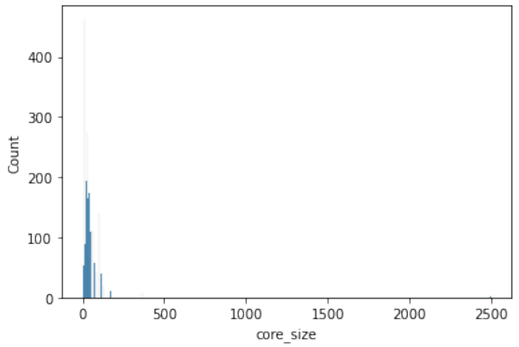
- **Распределение пропусков по признакам:**

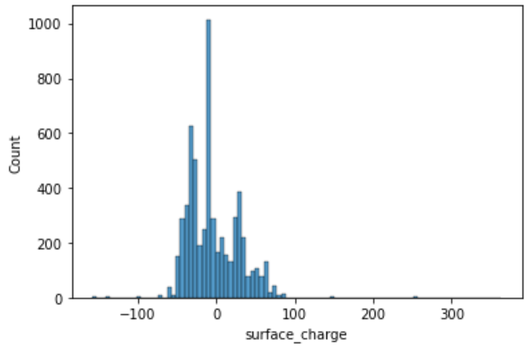


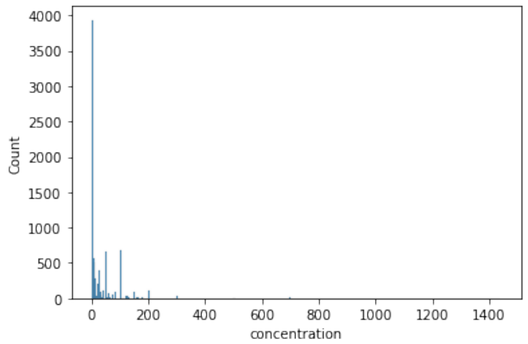
5. Анализ

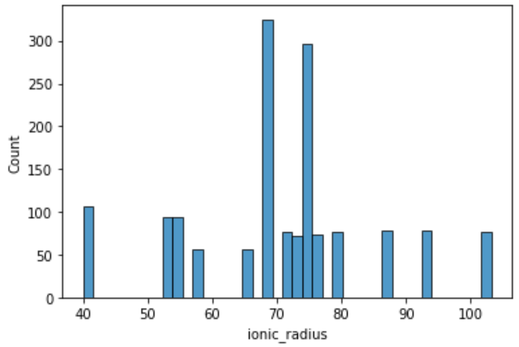
- **Распределение числовых признаков:**

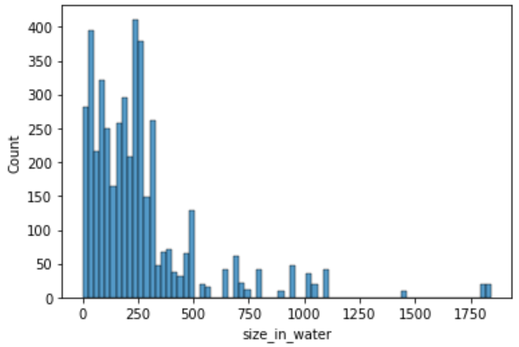


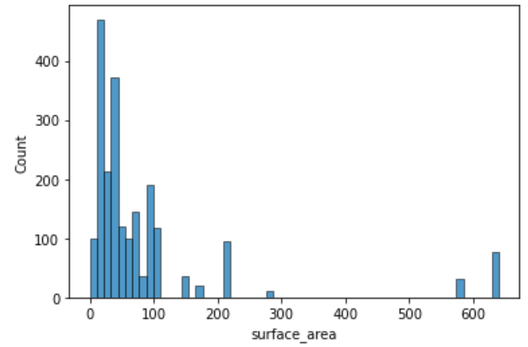


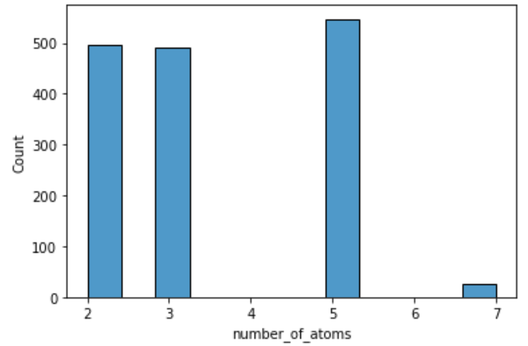


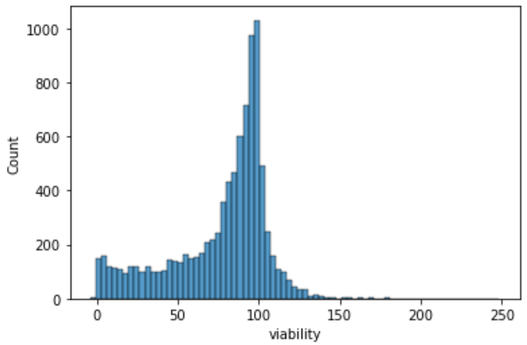


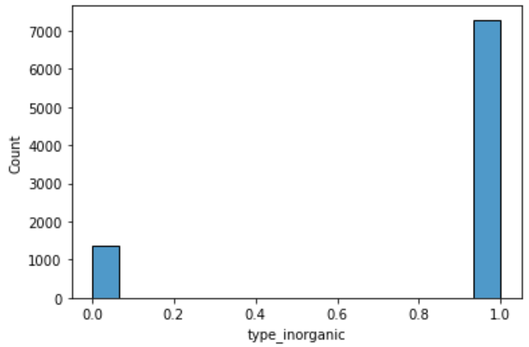


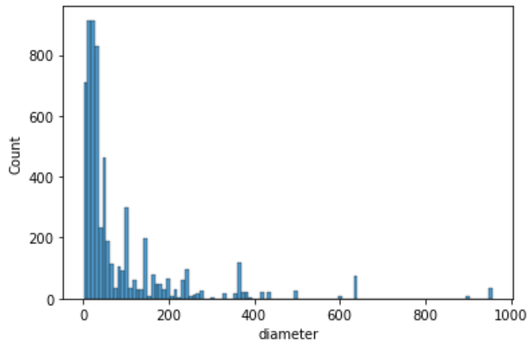


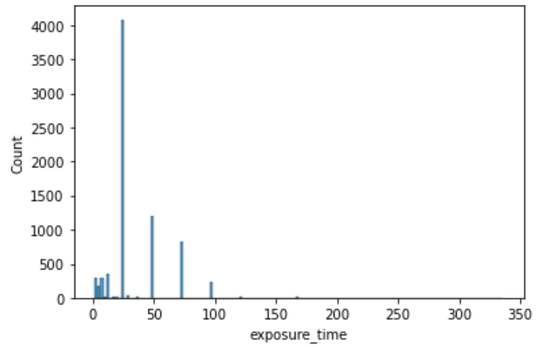






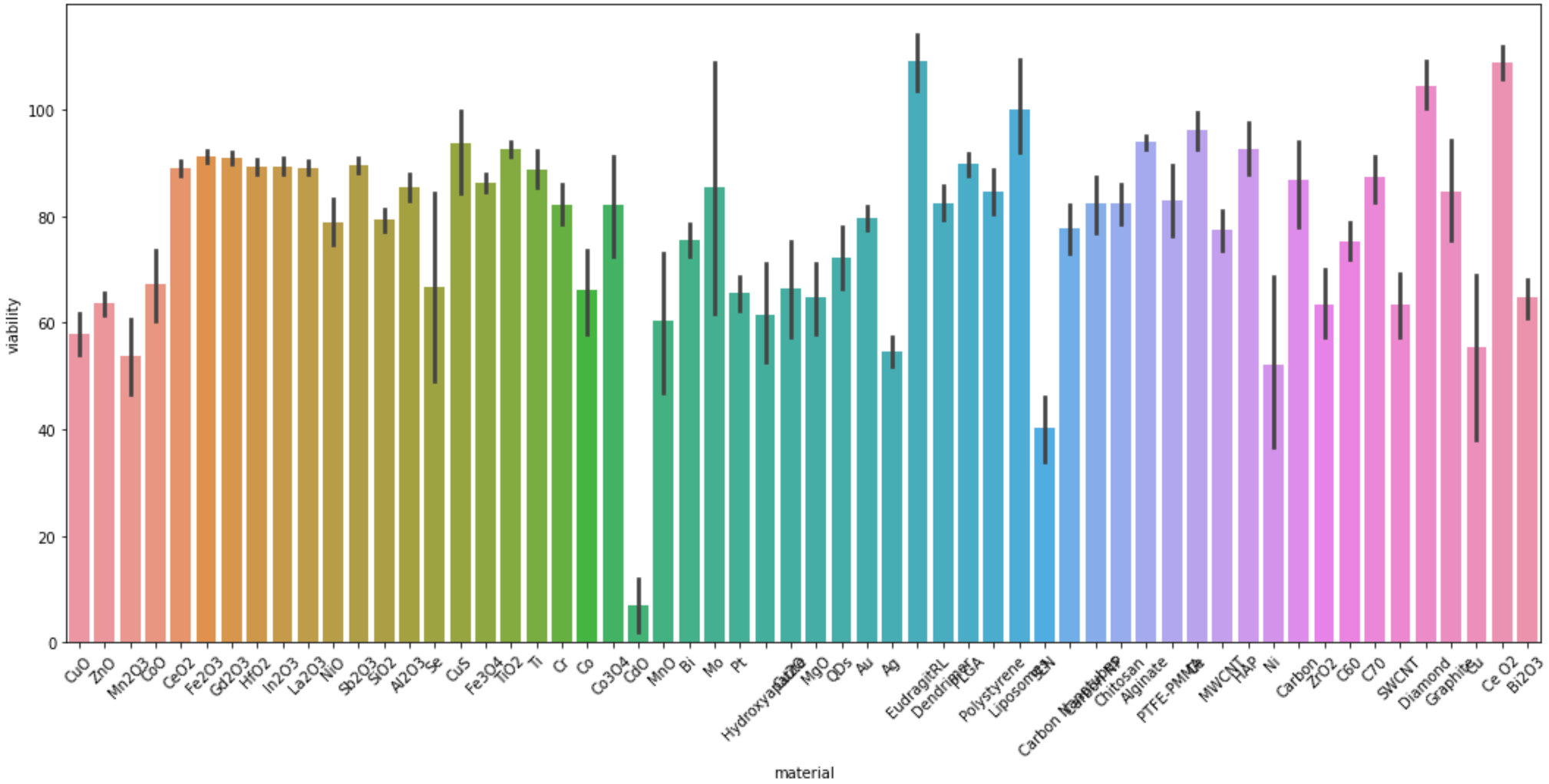






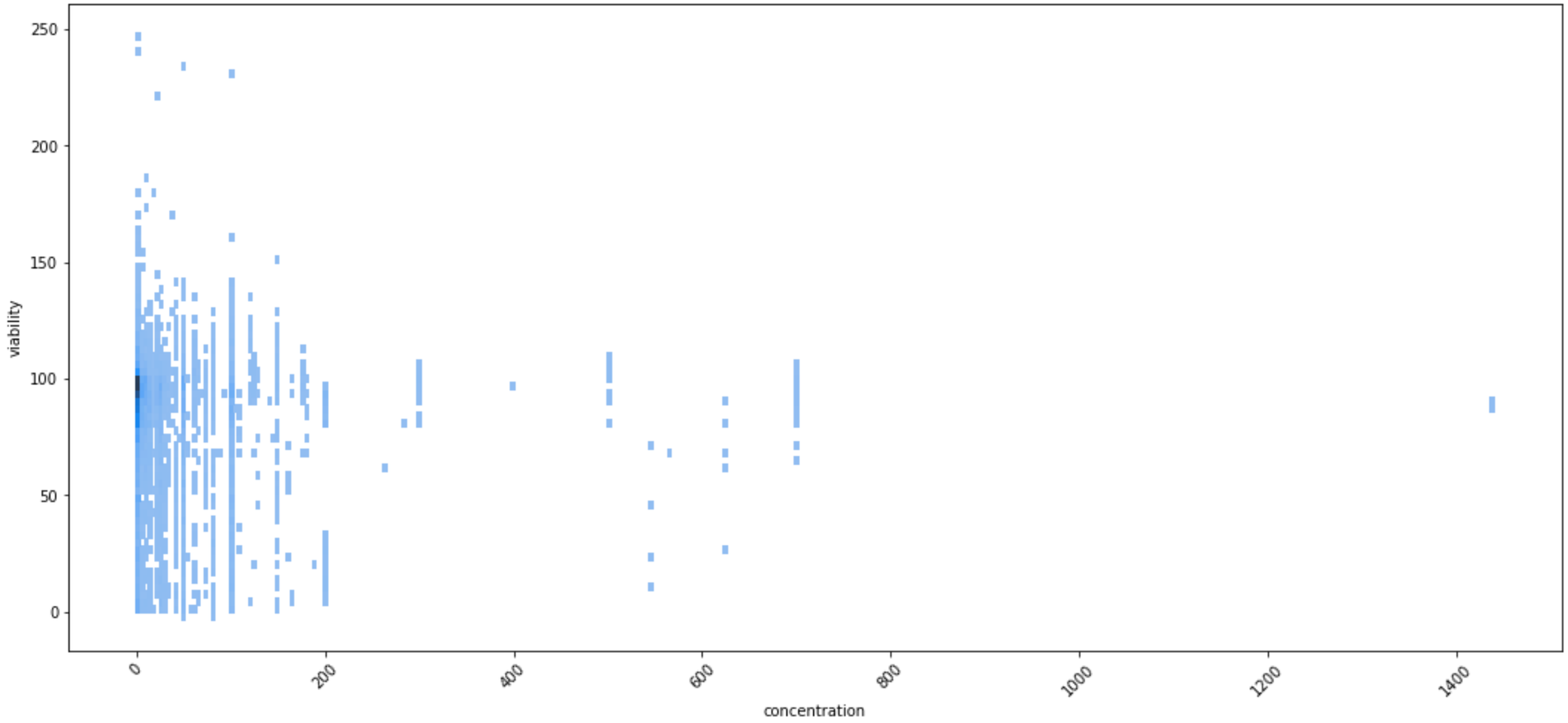
5. Анализ

- **Зависимость параметра *viability* от материала:**



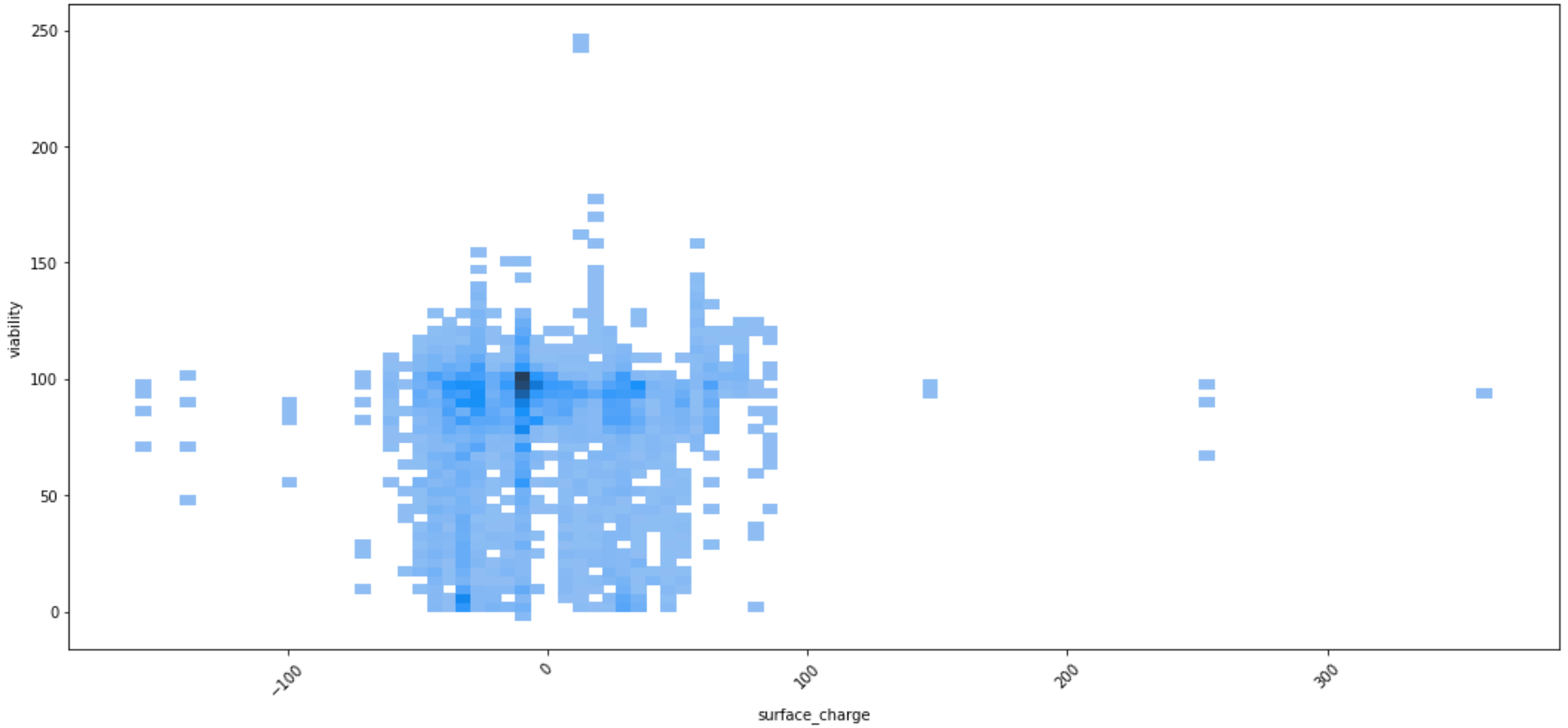
5. Анализ

- **Зависимость параметра *viability* от концентрации:**



5. Анализ

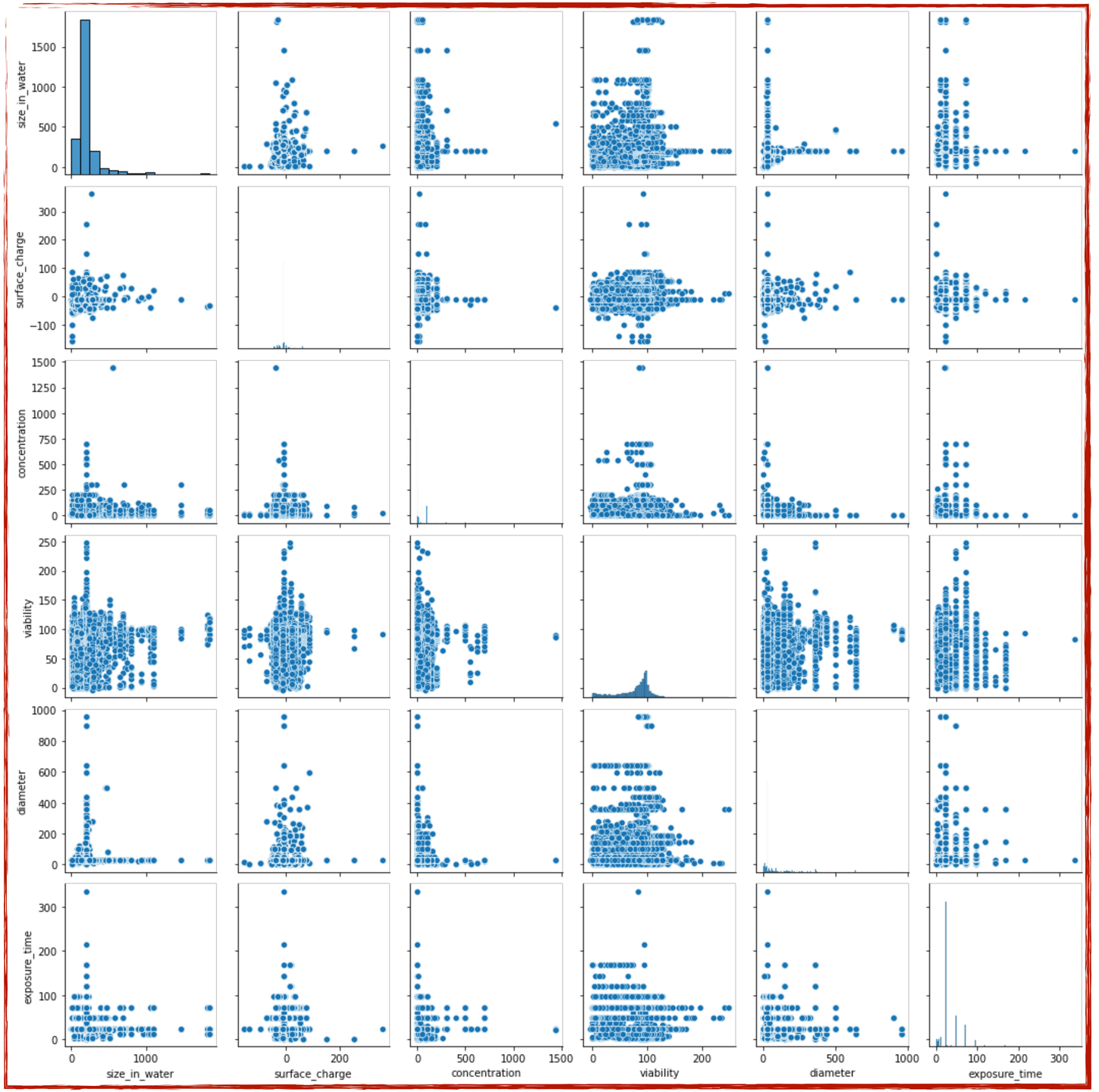
- **Зависимость параметра *viability* от заряда поверхности:**



5. Анализ

- **Корреляция признаков:** матрица графиков зависимости графиков друг от друга.

Прямолинейные зависимости и очевидные закономерности - отсутствуют



6. Модели

- **CatBoost**

```
%%time
estimator = CatBoostRegressor(random_seed = 42)
parameters = {
    'n_estimators': (200, 800, 100),
    'learning_rate': [0.1, 0.5, 0.1],
    'max_depth': [3, 6]
}


model_cb = GridSearchCV(estimator = estimator,
                        param_grid = parameters,
                        cv = 2, verbose=10)

# Обучение
model_cb.fit(features_train, target_train, cat_features=cat_features, verbose=2)
```

```
CatBoost
0.5462275143957516
{'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 800}
RMSE(CatBoost) =  19.19463126966019
```

GEEK BATTLE
DataCon
SCAMT

6. Модели

- **Random Forest Regressor**

```
estimator = RandomForestRegressor(random_state = 42)

param_grid = {
    'n_estimators': list(np.arange(50, 200, 20).astype(int)),
    'max_depth': list(np.arange(4, 20, 4).astype(int)),
}

model_rf = GridSearchCV(estimator = estimator,
                        param_grid = param_grid,
                        cv = 2, verbose=2)
# Обучение
model_rf.fit(features_train, target_train)
```

```
Random Forest Regressor
Лучшая метрика 0.525222403749226
полученная при использовании параметров: {'max_depth': 16, 'n_estimators': 190}
RMSE(RandomForest Regressor) = 20.941545331273932
```

GEEK BATTLE
DataCon
SCAMT

7. Вывод

- На основании данных о составе, свойствах, происхождении и тестах наночастиц из 5 таблиц нами построены 2 модели - **CatBoostRegressor** и **RandomForestRegressor**, предсказывающие признак viabillity.

- Была произведена предобработка данных: исправлены опечатки, удалены выбросы, обработаны пропуски. Базы данных объединены в сводную таблицу и сделана визуализация данных.

- Обе построенные модели показали близкие результаты: **RMSE(RandomForest Regressor) = 20.94, RMSE(CatBoost) = 19.19.**

GEEK BATTLE
DataCon
SCAMT