

Предсказание коэффициентов поглощения в сплавах

Вандышев Георгий (МО2-004и)

30 июня 2025 г.
МФТИ



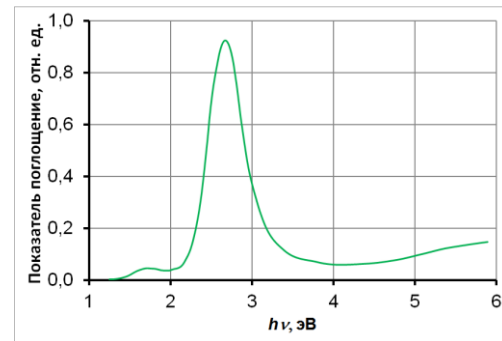
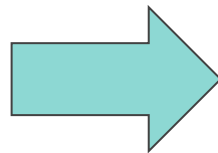
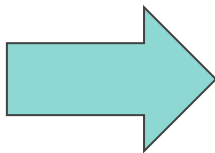
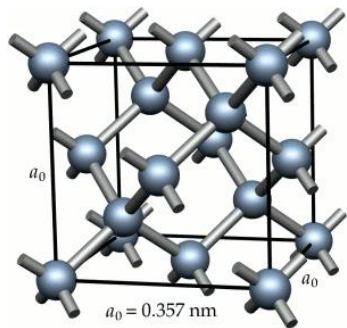


План

1. Задача
2. База данных
3. Первые шаги
4. Графовые нейросети
5. Еще GNN
6. Возвращение к истокам

Задача

Построение модели (ML), которая предсказывает по атомной структуре коэффициенты поглощения на длине волны 755 нм, 1064 нм и 1500 нм





База данных



1. По сплавам (видимо из Material Project + JARVIS)
2. Есть показатели поглощения k_{755} , k_{1064} , k_{1500}

```
1 id,structure_json,formula,spacegroup,bandgap,energy_above_hull,k_1064,k_755,k_1500
2 mp-546266,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[3.945744, 0.0, 0.0], [0.0, 3.945744, 0.0], [0.0, 0.0, 3.945744]]}},{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[3.87855986, 0.0, -1.1469436], [-0.339, 3.87855986, 0.0], [0.0, -0.339, 3.87855986]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
3 mp-9583,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[3.87855986, 0.0, -1.1469436], [-0.339, 3.87855986, 0.0], [0.0, -0.339, 3.87855986]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
4 mp-22988,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[3.87737825, -3.84681592, 0.01413623], [-3.84681592, 3.87737825, 0.01413623], [0.01413623, 0.01413623, 3.87737825]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
5 mp-1025029,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[-2.0265362, -3.51006494, 0.0], [-3.51006494, -2.0265362, 0.0], [0.0, 0.0, -2.0265362]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
6 mp-22867,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[4.2326573, -1e-08, 2.44372603], [-1e-08, 4.2326573, 2.44372603], [2.44372603, 2.44372603, 4.2326573]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
7 mp-1217120,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[6.908589, -2.104536, 0.0], [6.908589, 0.0, -2.104536], [-2.104536, 0.0, 6.908589]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
8 mp-3924,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[2.921784869999999, 0.0, -0.0], [-0.0, 2.921784869999999, 0.0], [0.0, -0.0, 2.921784869999999]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
9 mp-8181,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[3.57350791, -0.0, 0.0], [-1.78675396, 3.57350791, 0.0], [0.0, -1.78675396, 3.57350791]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
10 mp-13313,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[3.82366281, -0.01440131, 7.05734801], [-0.01440131, 3.82366281, 7.05734801], [7.05734801, 7.05734801, 3.82366281]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
11 mp-754326,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[3.16737726, 1.4935345500000001, -0.0], [1.4935345500000001, 3.16737726, 0.0], [-0.0, 0.0, 3.16737726]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
12 mp-23406,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[6.33114452, 1e-08, 3.65528831], [1e-08, 6.33114452, 3.65528831], [3.65528831, 3.65528831, 6.33114452]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
13 mp-9564,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[4.35434578, -1e-08, -0.0], [-1e-08, 4.35434578, -0.0], [-0.0, -0.0, 4.35434578]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
14 mp-2691,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[3.76029987, 1e-08, 2.17100981], [1e-08, 3.76029987, 2.17100981], [2.17100981, 2.17100981, 3.76029987]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
15 mp-856,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[3.20749977, 0.0, 0.0], [0.0, 3.20749977, 0.0], [0.0, 0.0, 3.20749977]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
16 mp-30530,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[4.61161841, -0.0, 2.66251818], [-0.0, 4.61161841, 2.66251818], [2.66251818, 2.66251818, 4.61161841]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
17 mp-985586,"{'@module': 'pymatgen.core.structure', '@class': 'Structure', 'charge': 0, 'lattice': {'matrix': [[4.53313, 0.0, 0.0], [0.0, 4.53313, 0.0], [0.0, 0.0, 4.53313]]}}",Fe2O3,225,0.0,0.0,0.0,0.0,0.0
```



База данных full_mp+jv_5k_stable_bg_dataset.csv

Надо изменить по формату JVASP

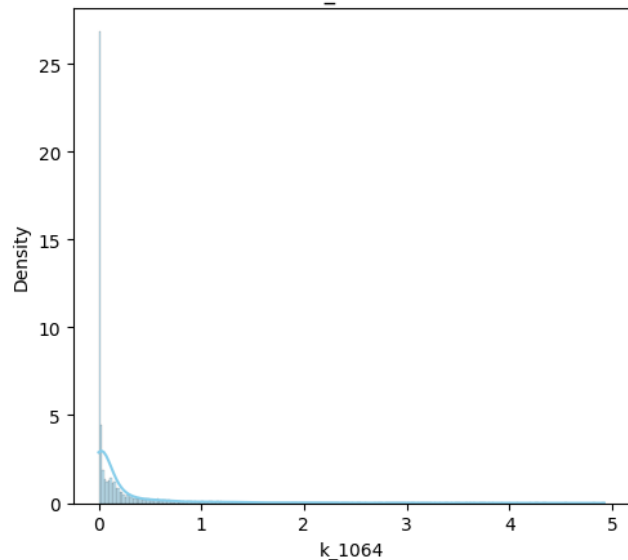
```
872 mp-12671,"{'@module': 'pymatgen.core.structure', '@class':  
873 mp-1065514,"{'@module': 'pymatgen.core.structure', '@clas  
874 mp-7439,"{'@module': 'pymatgen.core.structure', '@class':  
875 mp-13276,"{'@module': 'pymatgen.core.structure', '@class'  
876 mp-999138,"{'@module': 'pymatgen.core.structure', '@class'  
877 JVASP-10004,"{'"@module"": 'pymatgen.core.structure'", "  
878 JVASP-10005,"{'"@module"": 'pymatgen.core.structure'", "  
879 JVASP-10006,"{'"@module"": 'pymatgen.core.structure'", "  
880 JVASP-100061,"{'"@module"": 'pymatgen.core.structure'", "  
881 JVASP-100066,"{'"@module"": 'pymatgen.core.structure'", "  
882 JVASP-100068,"{'"@module"": 'pymatgen.core.structure'", "  
883 JVASP-100082,"{'"@module"": 'pymatgen.core.struc
```

```
872 3.73803603, 'b': 3.7380362172317976, 'c': 6.52200321, 'alpha': 90.0, 'beta': 90.0  
873 'pbc': (True, True, True), 'a': 8.530465155454591, 'b': 8.530465155454591, 'c': 8  
874 02, 7.41974626]], 'pbc': (True, True, True), 'a': 7.41985541727002, 'b': 7.4198558  
875 ': 4.37129639, 'b': 4.371296435707485, 'c': 8.04126415, 'alpha': 90.0, 'beta': 90.  
876 , True, True), 'a': 4.113385950241802, 'b': 4.11338546, 'c': 7.849834247179252, 'a  
877 -2.048668932051024], [-0.0, -0.0, 6.146006796153073]], '"pbc"' : [true, true, true  
878 317, -1.3190121126540906], [-0.0182141080279125, -0.0266613273390963, 7.2050021596  
879 379669327182], [2.2138603e-09, -0.0099599470688208, 7.460534963042954]], '"pbc"' :  
880 17967415, -3.128010863482596], [-0.0410859758335717, 0.0223294333571995, 6.0315319  
881 02009, -0.4795040837361508], [0.0285644983335913, -0.008199259737792, 5.4435893607  
882 2.6885406476151035], [0.0, -0.0, 5.377081295230207]], '"pbc"' : [true, true, true],  
883 3.8536110643851966], [1.01351681e-08, 7.1666453e-09, 7.7072230936207005]], '"pbc"  
884 26922, -1.577419104477233], [-0.0081277243382029, 0.0111210833438285, 5.883532694
```

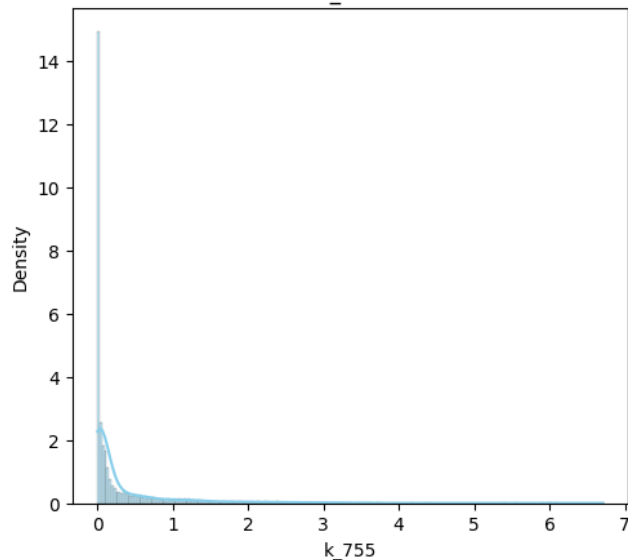


Распределение сплавов по k

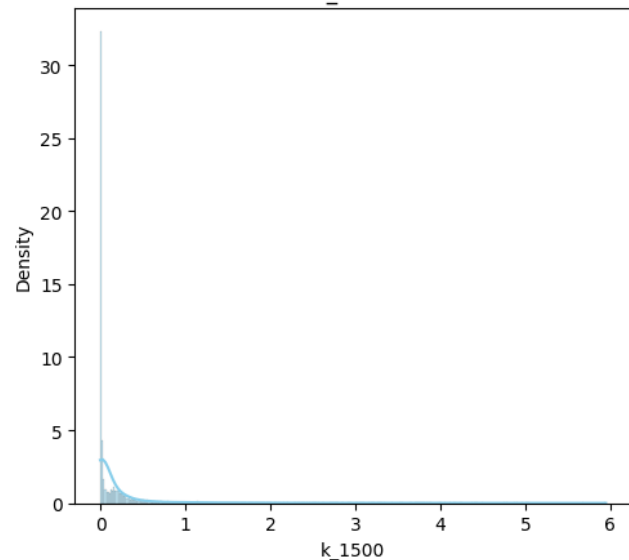
Исходное распределение
k_1064



Исходное распределение
k_755



Исходное распределение
k_1500





Первые шаги

`RandomForest.ipynb`





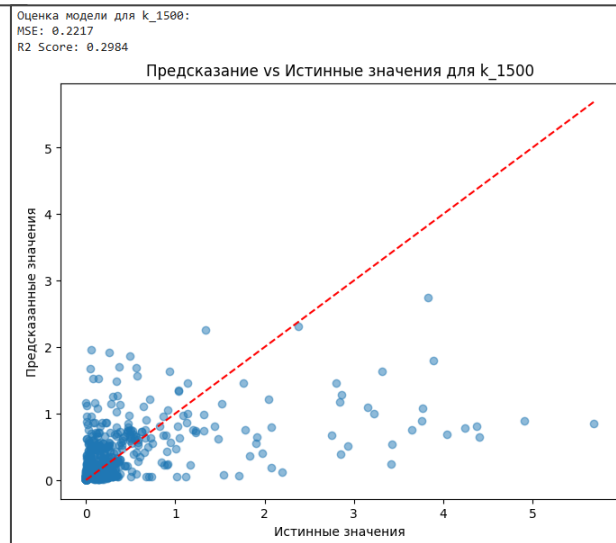
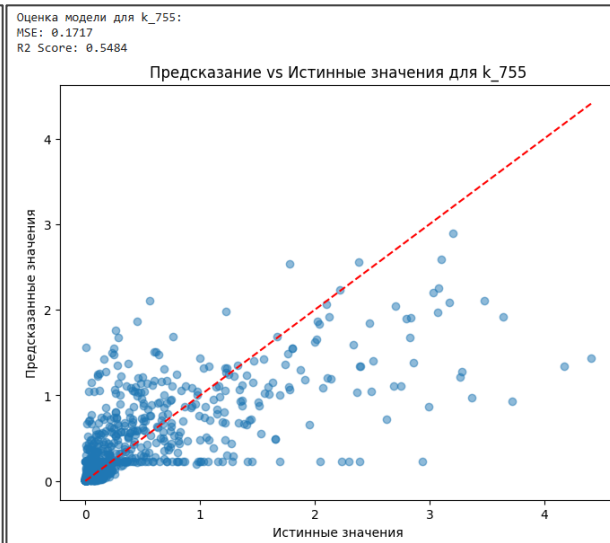
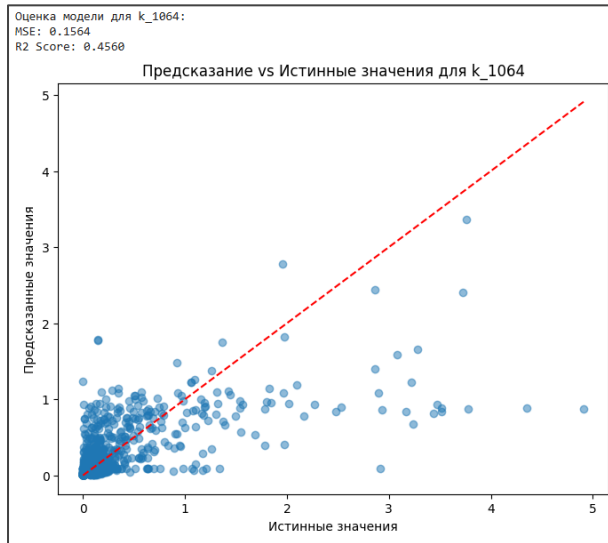
Первые шаги

1. В начале было решено протестировать классическую ML модель - RandomForest
2. Из таблицы “full_mp+jv_5k_stable_bg_dataset.csv” были получены свойства сплавов (базовые)
3. Была обучена модель в библиотеке sklearn для каждого значения k
4. Были исследованы гиперпараметры модели -> получены наилучшие

```
feature_names = [  
    'volume', 'bandgap', 'energy_above_hull',  
    'a', 'b', 'c', 'alpha', 'beta', 'gamma',  
    'num_elements', 'spacegroup'  
]
```



Результаты для RandomForest



Все не очень хорошо
Пример вывода GridSearch:

```
{'max_depth': 11, 'max_leaf_nodes': 250, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 500} -0.19925332709743784
```

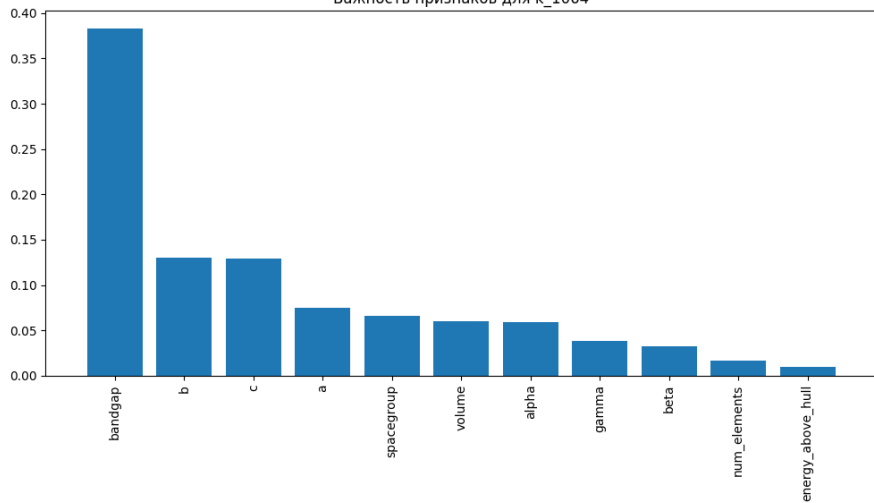


Важность признаков

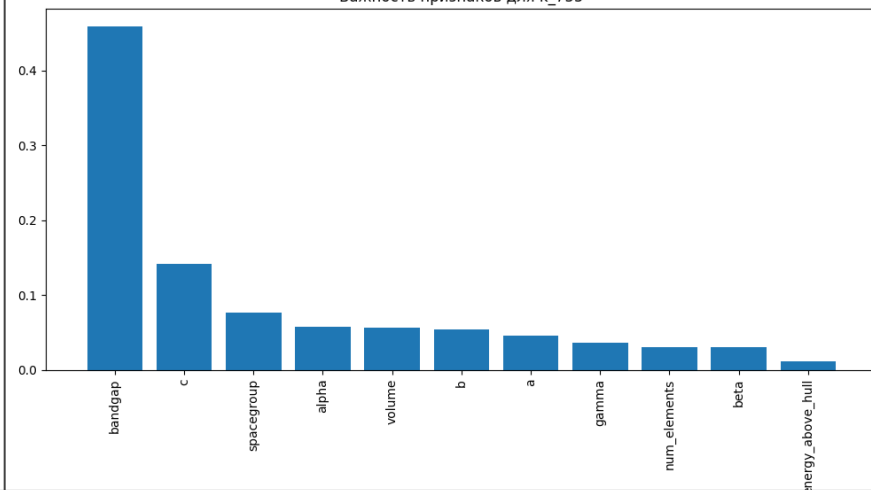
Самый важный признак - band gap

Далее параметры кристаллической решетки

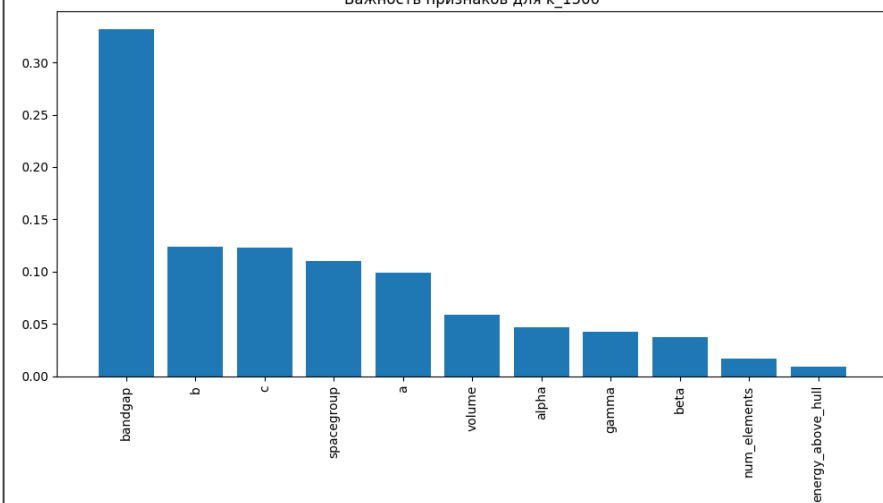
Важность признаков для k_1064



Важность признаков для k_755



Важность признаков для k_1500



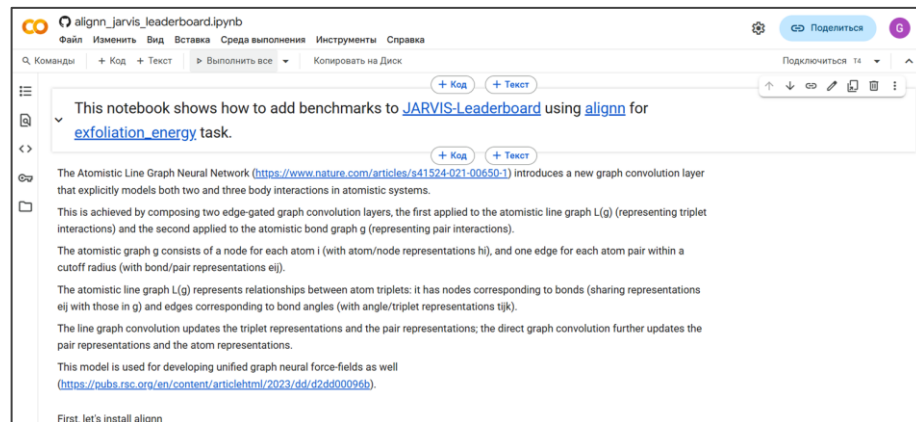
Графовые нейросети

Data_forALIGNN.ipynb + ALIGNN.ipynb



Графовые нейросети

1. Выбрана ALIGNN
2. Найден ноутбук в качестве примера (официальный - git)
3. Входные параметры = POSCAR файлы VASP
4. Перевод таблицы в папку с входными файлами (alignn/alignn_data)
5. config_k.json - config файл чуть-чуть изменен



https://colab.research.google.com/github/knc6/jarvis-tools-notebooks/blob/master/jarvis-tools-notebooks/alignn_jarvis_leaderboard.ipynb



Проблемы

1. Не полная документация
2. Использует не все конфигурации (через раз)

```
data range 4.92489910736036 4.5216015119897497e-05  
line_graph True  
100% 6868/6868 [02:41<00:00, 42.40it/s]  
data range 3.720222773061368 0.0  
line_graph True  
100% 858/858 [00:24<00:00, 34.63it/s]  
data range 2.9435545702461887 0.0003154817802712  
line_graph True  
100% 858/858 [00:18<00:00, 45.65it/s]  
n_train: 6868  
n_val : 858  
n_test : 858  
rank 0  
world_size 1
```

Использует все структуры

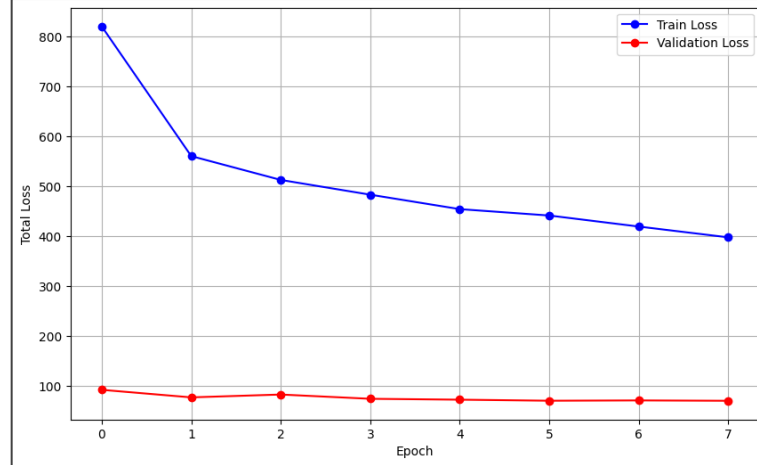
```
data range 6.7109952669829545 4.348977225287308e-05  
line_graph True  
100% 4768/4768 [00:00<00:00, 2807981.11it/s]  
Reading dataset Atrain_data  
data range 4.408488083342374 0.0  
line_graph True  
100% 596/596 [00:00<00:00, 2334085.14it/s]  
Reading dataset Aval_data  
data range 4.8731407918931 0.000225536621609  
line_graph True  
100% 596/596 [00:00<00:00, 2537873.28it/s]  
Reading dataset Atest_data  
n_train: 50  
n_val : 6  
n_test : 6  
rank 0
```

Использует не все структуры (очень мало)
Почему?

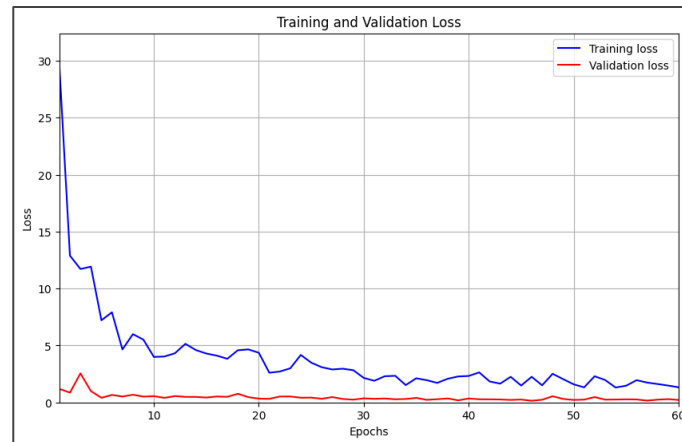


Проблемы

1. Не полная документация
2. Использует не все конфигурации (через раз)
3. Непонятно почему Loss для val такой маленький



Нормальное обучение



Не нормальное обучение (на малом количестве данных)



Проблемы

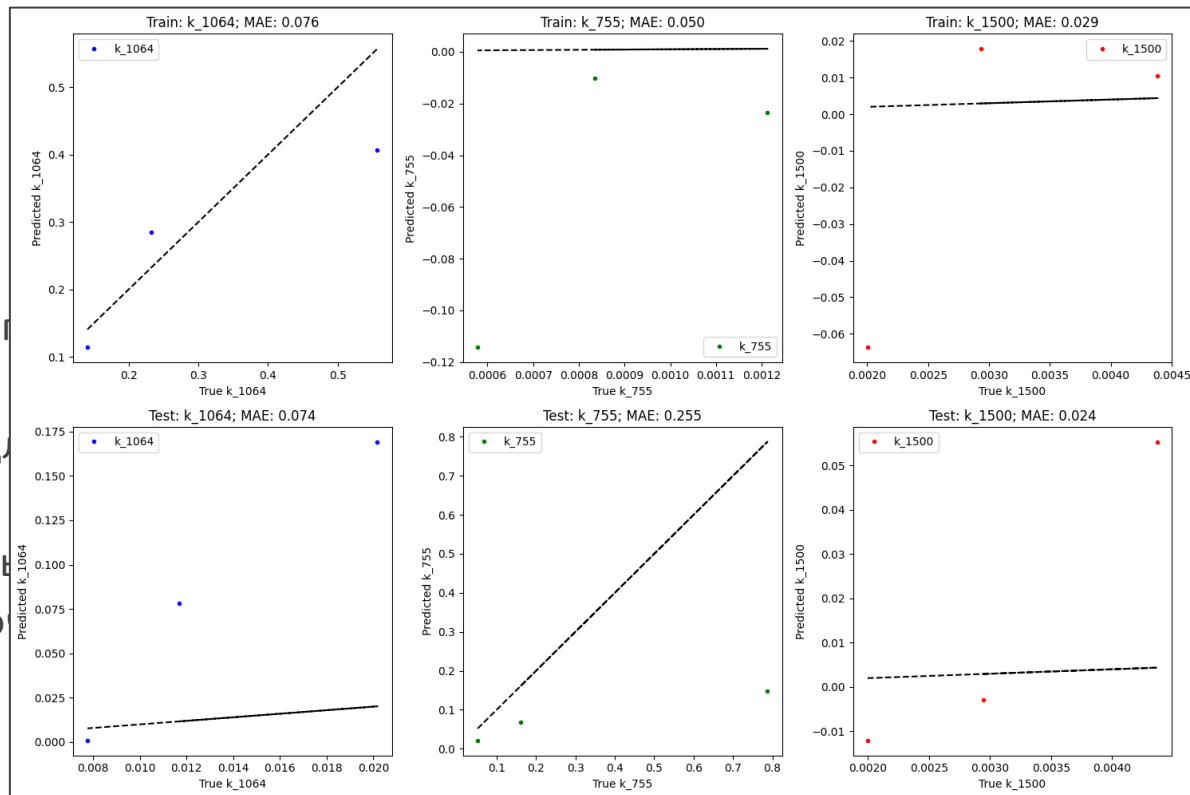
1. Не полная документация
2. Использует не все конфигурации (через раз)
3. Непонятно почему Loss для val такой маленький
4. В качестве результатов выводит случайное количество точек (не разобрался до конца)

```
Overall MAE:
k_1064:
  Train MAE: 0.0764
  num points: 3
  Test MAE: 0.0741
  num points: 3
k_755:
  Train MAE: 0.0502
  num points: 3
  Test MAE: 0.2548
  num points: 3
k_1500:
  Train MAE: 0.0289
  num points: 3
  Test MAE: 0.0236
  num points: 3
```




Проблемы

1. Не полная документация
2. Использует не все конфигурации (через раз)
3. Непонятно почему Loss для маленького
4. В качестве результатов выдает случайное количество точек (не разобрался до конца)



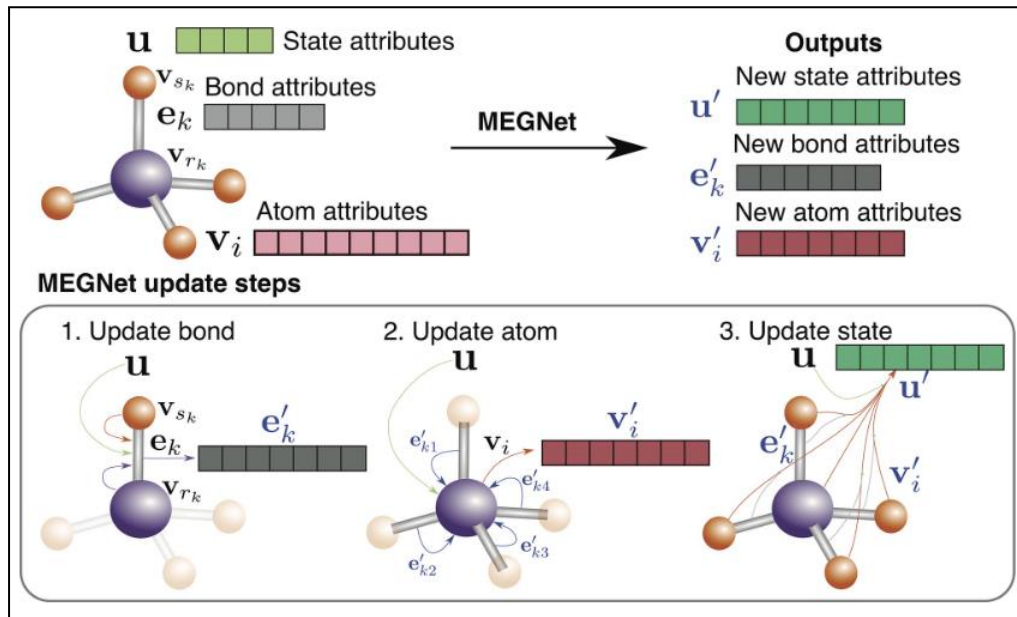
Еще GNN (Проблемы продолжаются)

`crystal_example.ipynb`



MegNet

1. Решено использовать другую модель





1. Решено использовать другую модель
2. Найден ноутбук в качестве примера (официальный - git)

co

crystal_example.ipynb

☆ ☁

ФайлИзменитьВидВставкаСреда выполненияИнструментыСправка

🔍 Команды+ Код+ Текст▶ Выполнить все ▾

☰
🔍
⏮
🔑
📄

⌵ Load data

```
[ ] | pip install monty megnet

import numpy as np
from monty.json import MontyDecoder
from monty.serialization import loadfn

data = loadfn('bulk_moduli.json')
structures = data['structures']
targets = np.log10(data['bulk_moduli'])

[ ] | len(structures)

100
```

⌵ Set up model and train

```
[ ] | from megnet.models import MEGNetModel
from megnet.data.crystal import CrystalGraph
import numpy as np

nfeat_bond = 10
rcutoff = 5
```

<https://colab.research.google.com/drive/1CHAsLh0klj5K82f8ZOcK9pLfV9Bx4OgK#scrollTo=A-LIUEDKlq1F>



MegNet

1. Решено и
2. Найден н
(официал

crystal_example.ipynb

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка

Q Команды + Код + Текст ▶ Выполнить все

Set up model and train

```
from megnet.models import MEGNetModel
from megnet.data.crystal import CrystalGraph
import numpy as np

nfeat_bond = 10
r_cutoff = 5
gaussian_centers = np.linspace(0, r_cutoff + 1, nfeat_bond)
gaussian_width = 0.5
graph_converter = CrystalGraph(cutoff=r_cutoff)
model = MEGNetModel(graph_converter=graph_converter, centers=gaussian_centers, width=gaussian_width)
```

Traceback (most recent call last)
<ipython-input-11-342210229> in <cell line: 0>()
8 gaussian_width = 0.5
9 graph_converter = CrystalGraph(cutoff=r_cutoff)
--> 10 model = MEGNetModel(graph_converter=graph_converter, centers=gaussian_centers, width=gaussian_width)

↕ 2 frames

/usr/local/lib/python3.11/dist-packages/keras/src/utils/tracking.py in wrapper(*args, **kwargs)
24 def wrapper(*args, **kwargs):
25 with DotNotTrackScope():
--> 26 return fn(*args, **kwargs)
27
28 return wrapper

TypeError: Trainer.compile() got an unexpected keyword argument 'sample_weight_mode'

Не работает тестовый ноутбук?!



Возврат к истокам

MultipleFeaturizerRegressor.ipynb



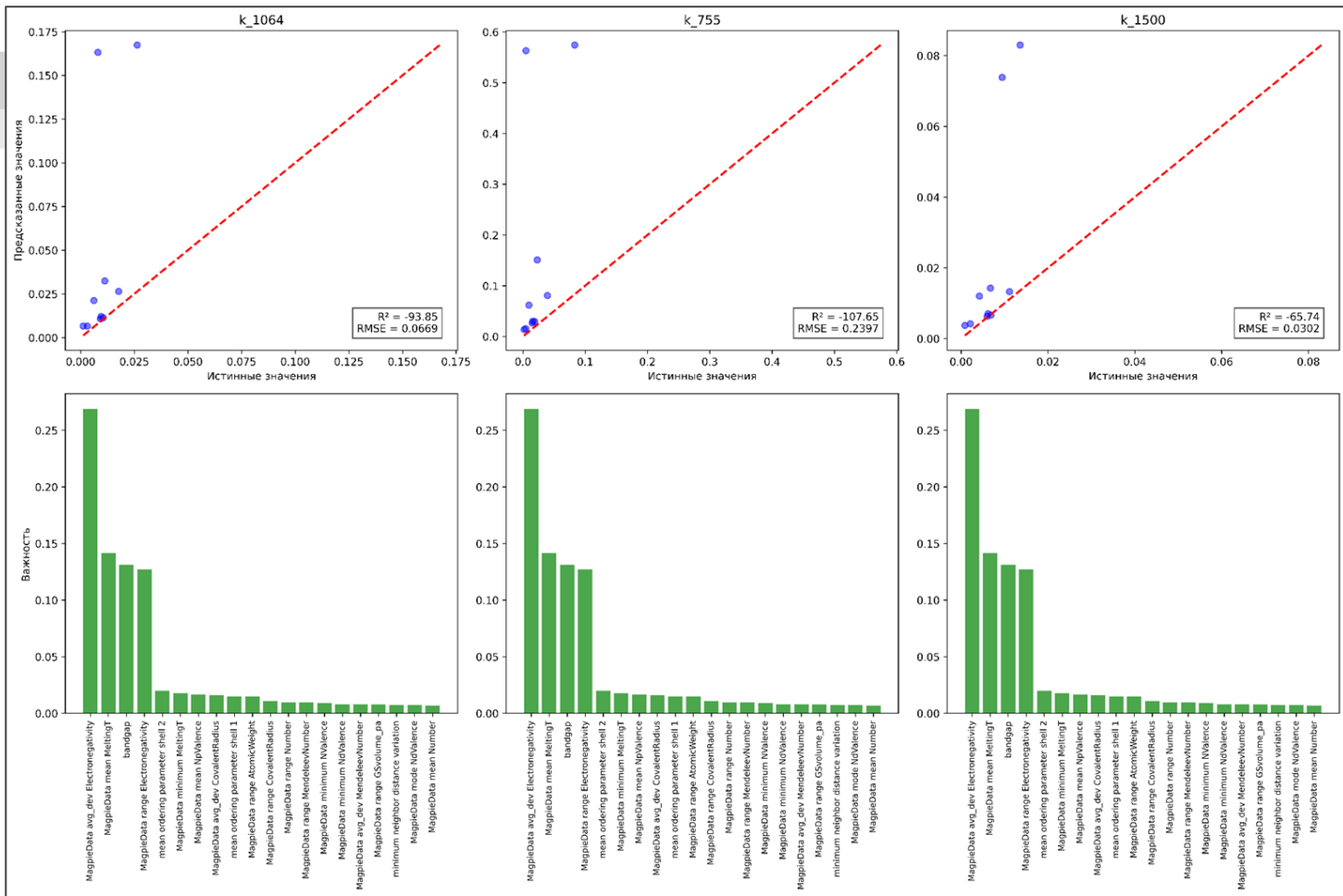


Структурные дескрипторы для классических регрессоров

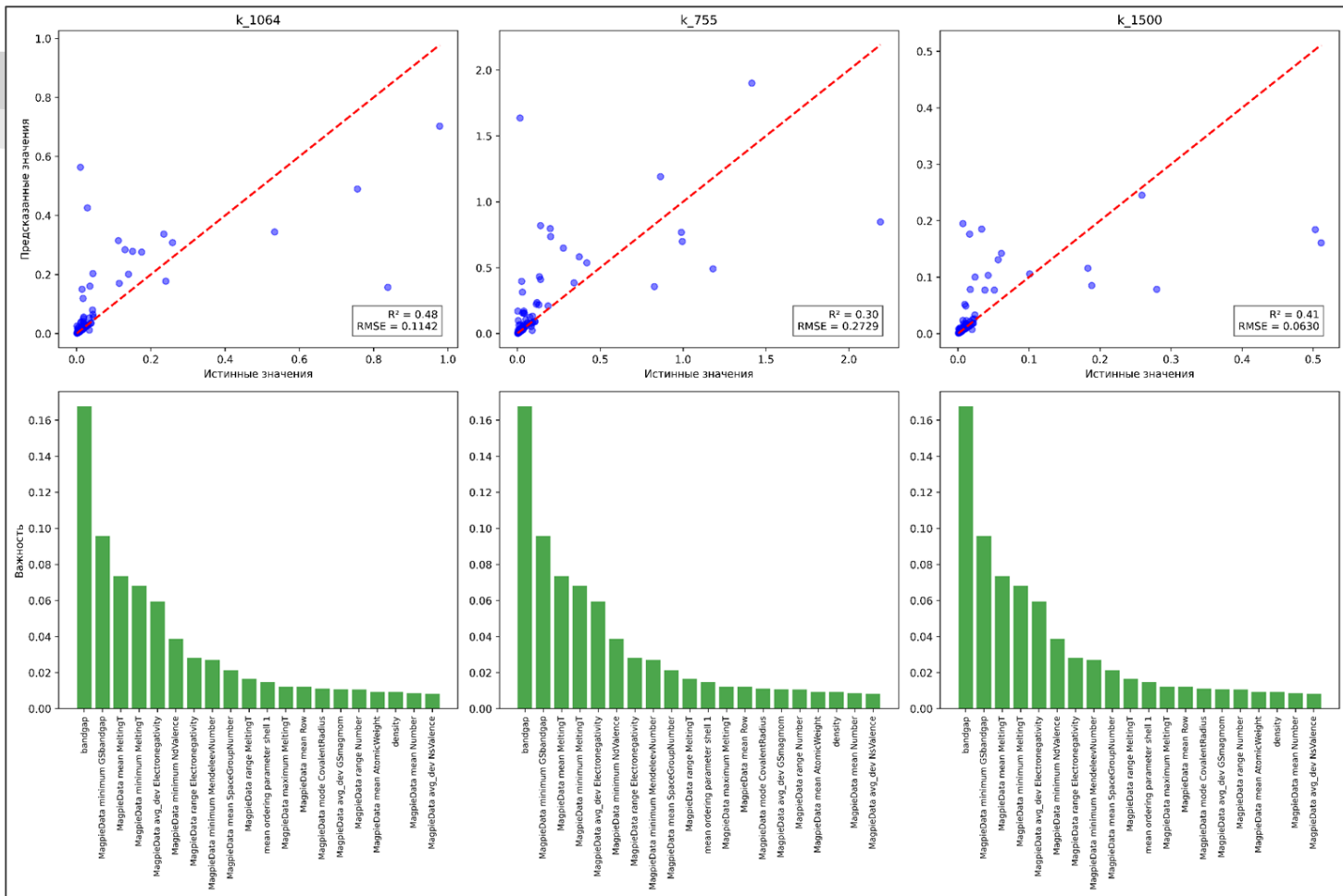
```
# Извлечения признаков из структуры
featurizer = MultipleFeaturizer([
    SiteStatsFingerprint.from_preset("CoordinationNumber_ward-prb-2017"),
    StructuralHeterogeneity(),
    ChemicalOrdering(),
    StructureComposition(ElementProperty.from_preset("magpie")),
    DensityFeatures()
])

X_features = featurizer.featurize_many(data["structure"], pbar=False, ignore_errors=True )
X_features = pd.DataFrame(X_features, columns=featurizer.feature_labels())
```

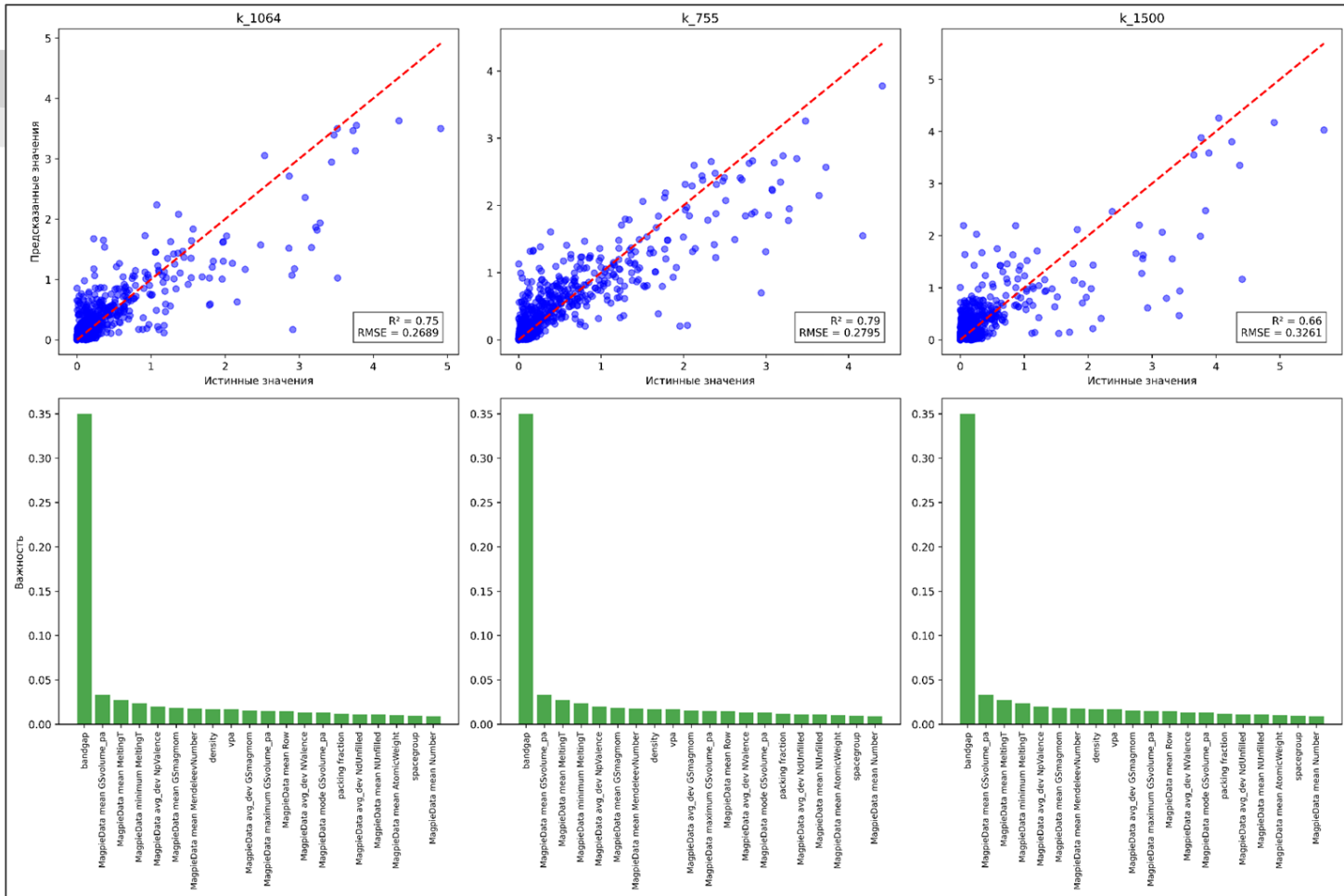
Результаты для разных размеров выборок (N=50)



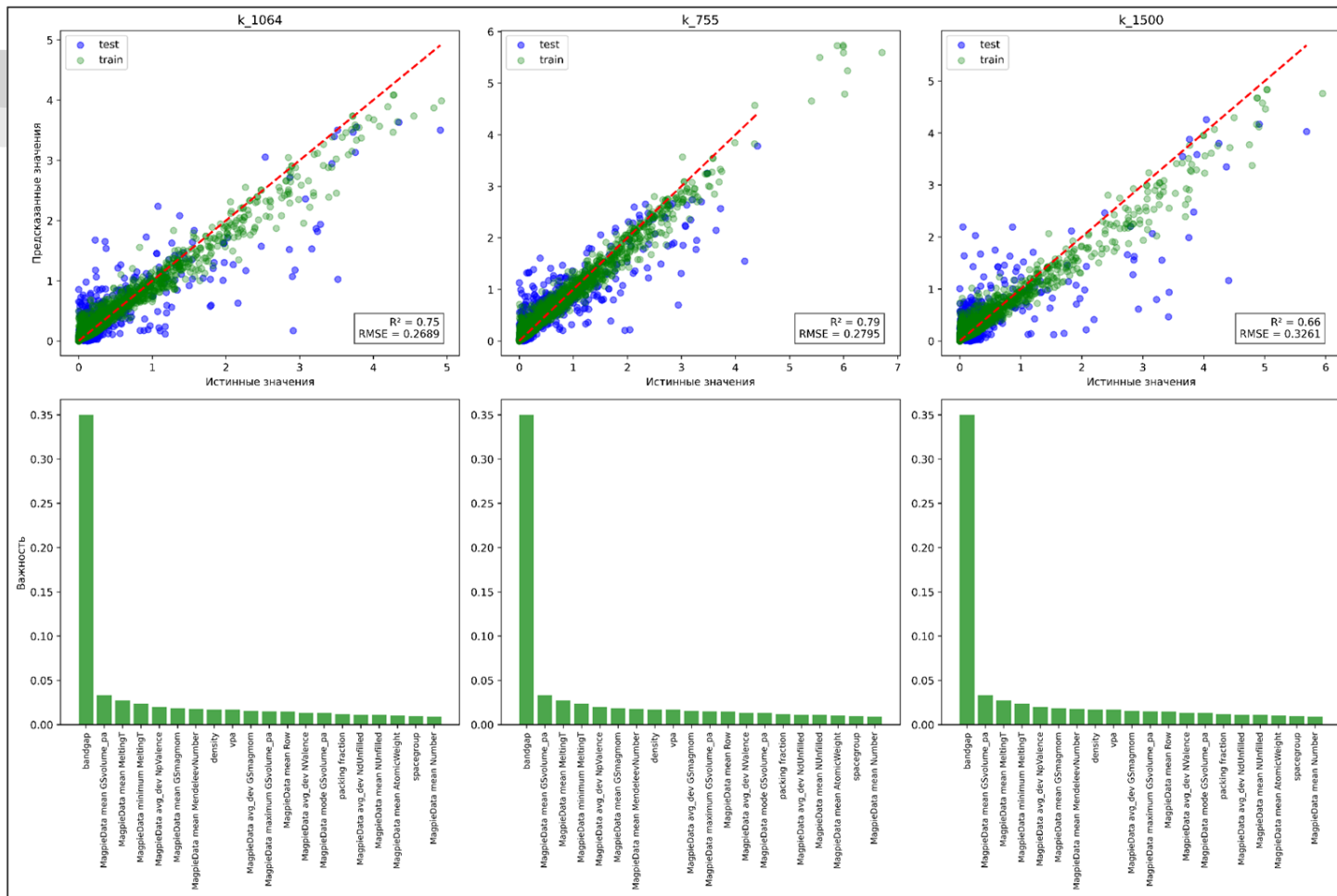
Результаты для разных размеров выборок (N=500)



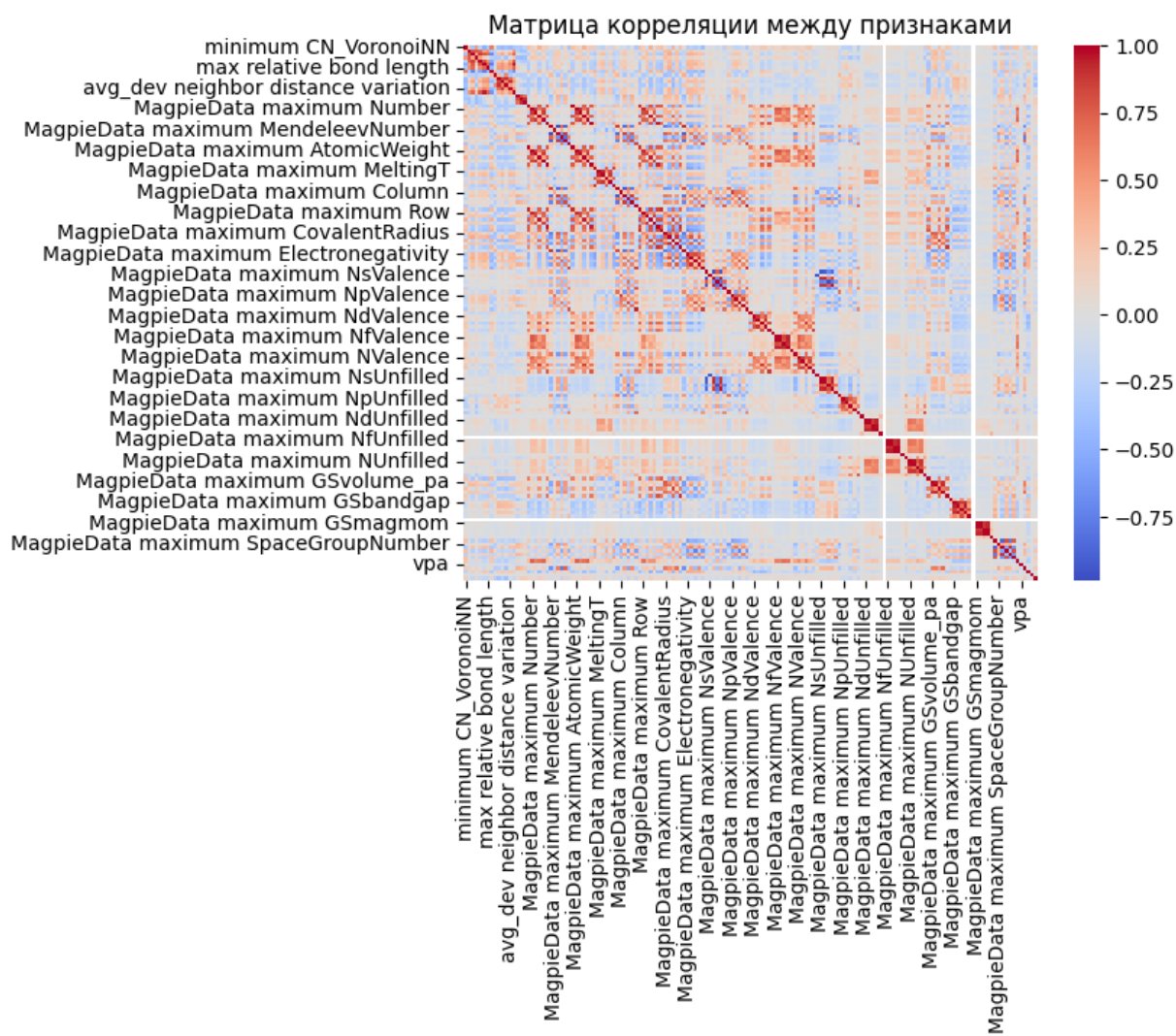
Результаты для разных размеров выборок (N=5960)



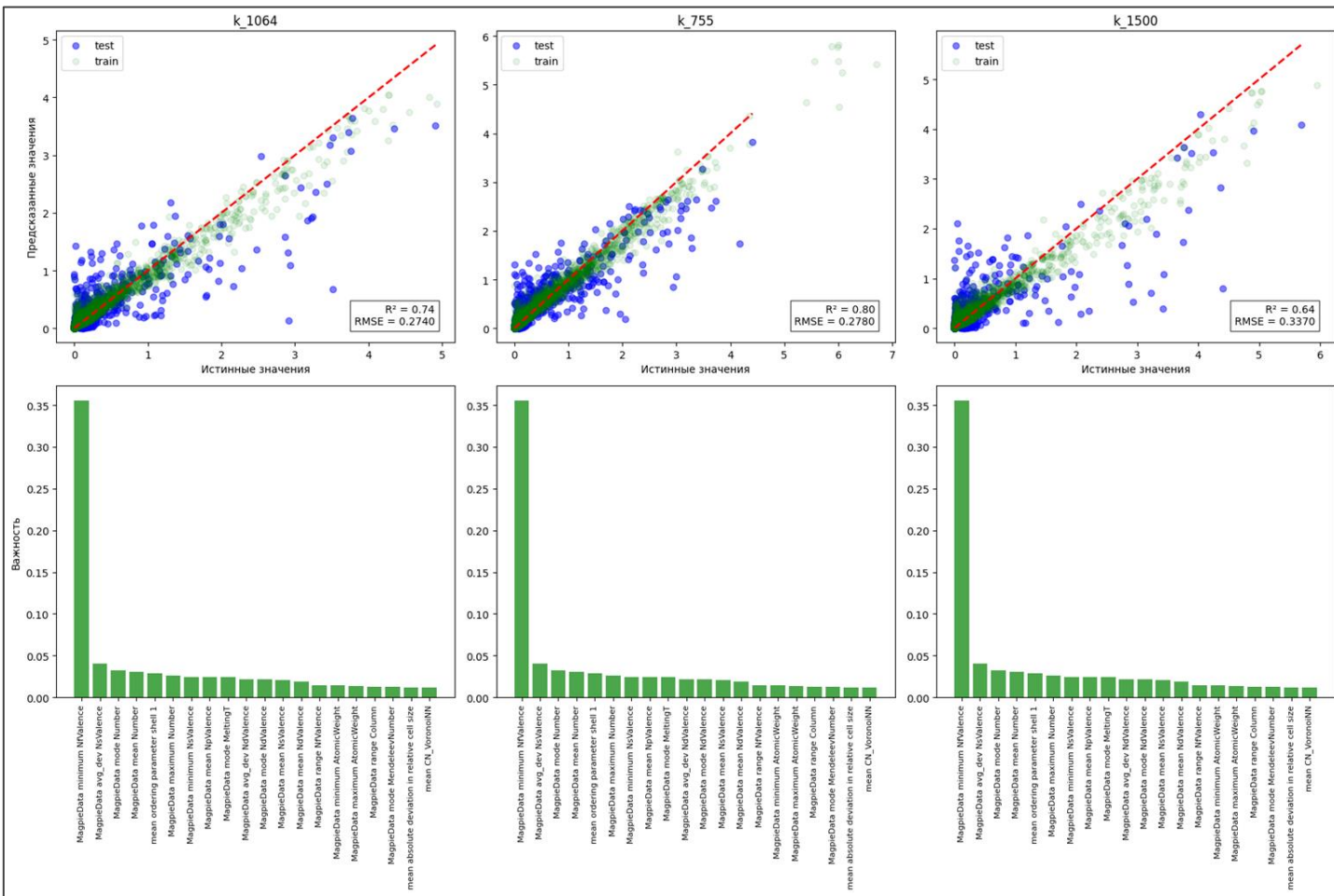
Результаты для GridSearch



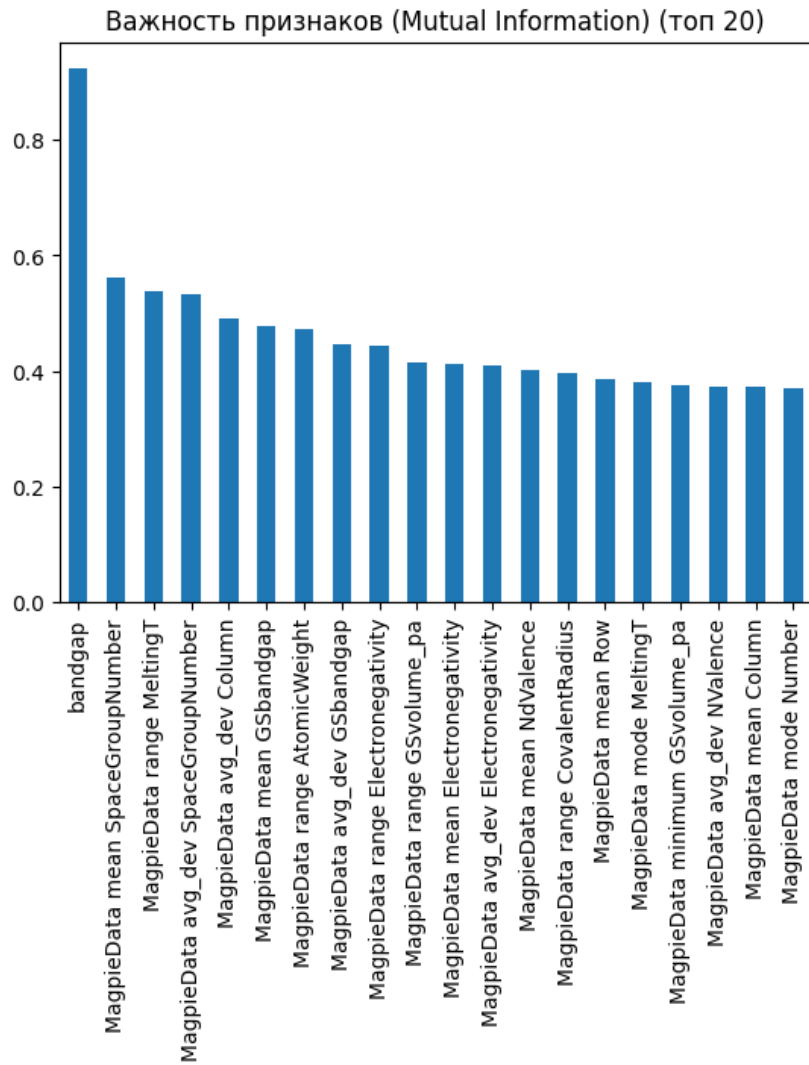
Корреляции



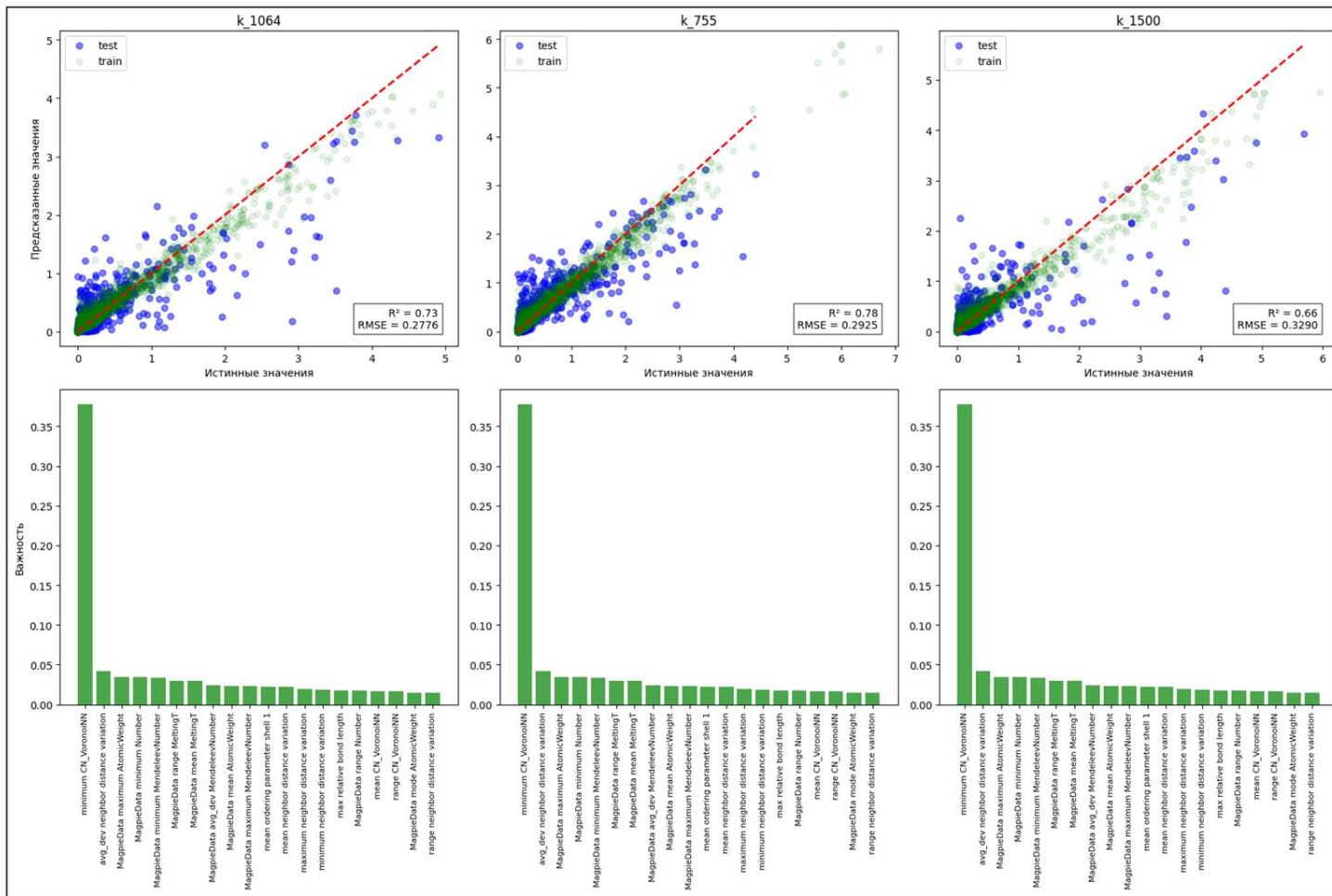
Результаты после удаления по корреляции



Нелинейный анализ



Результаты для топ40 по MI





Итоги

1. Обучено несколько моделей
2. На оптические свойства больше всего влияет band gap



@pechenkapop



vandyshev.gk@phystech.edu