Python与机器学习

——机器学习经典案例2

华算科技 黄老师

2022年1月20日

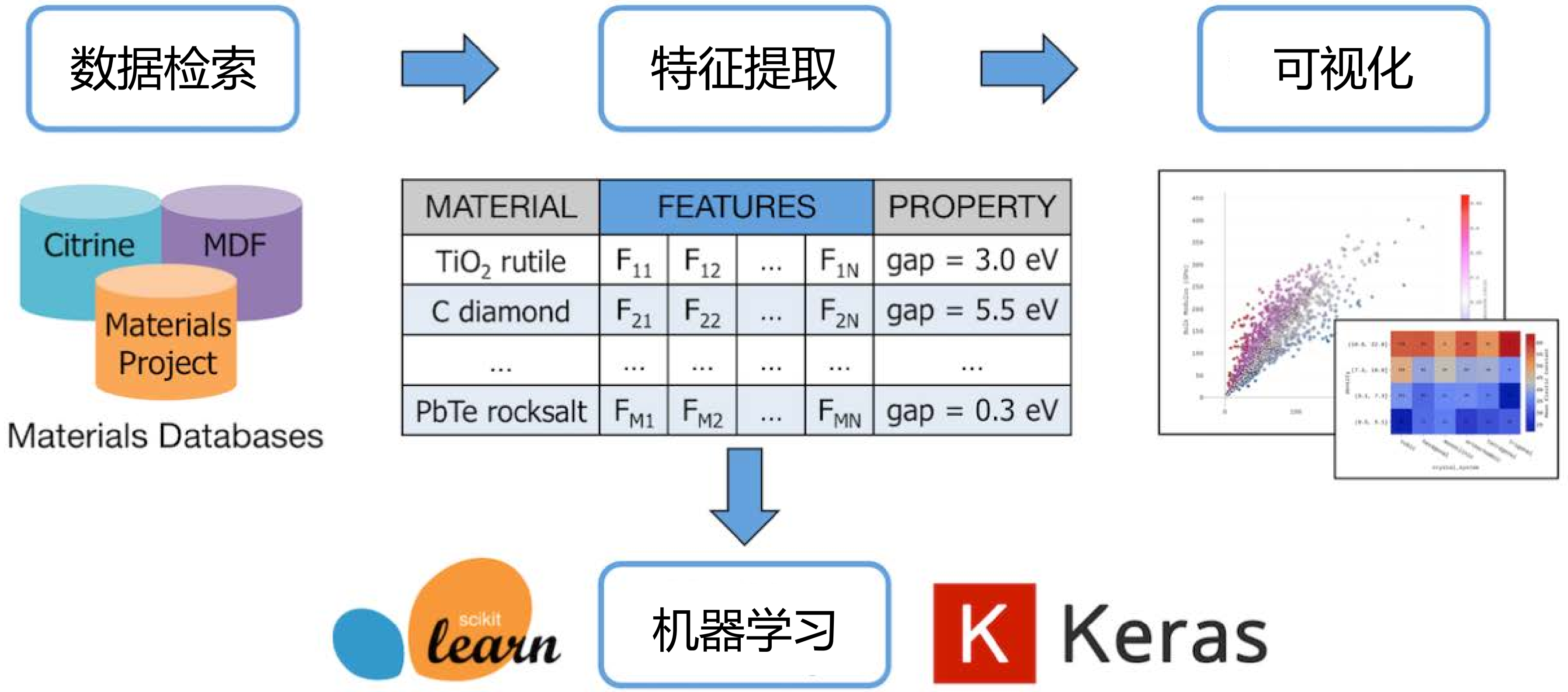
华算科技
HUASUAN TECHNOLOGY

目录

matminer

matminer

https://hackingmaterials.lbl.gov/matminer



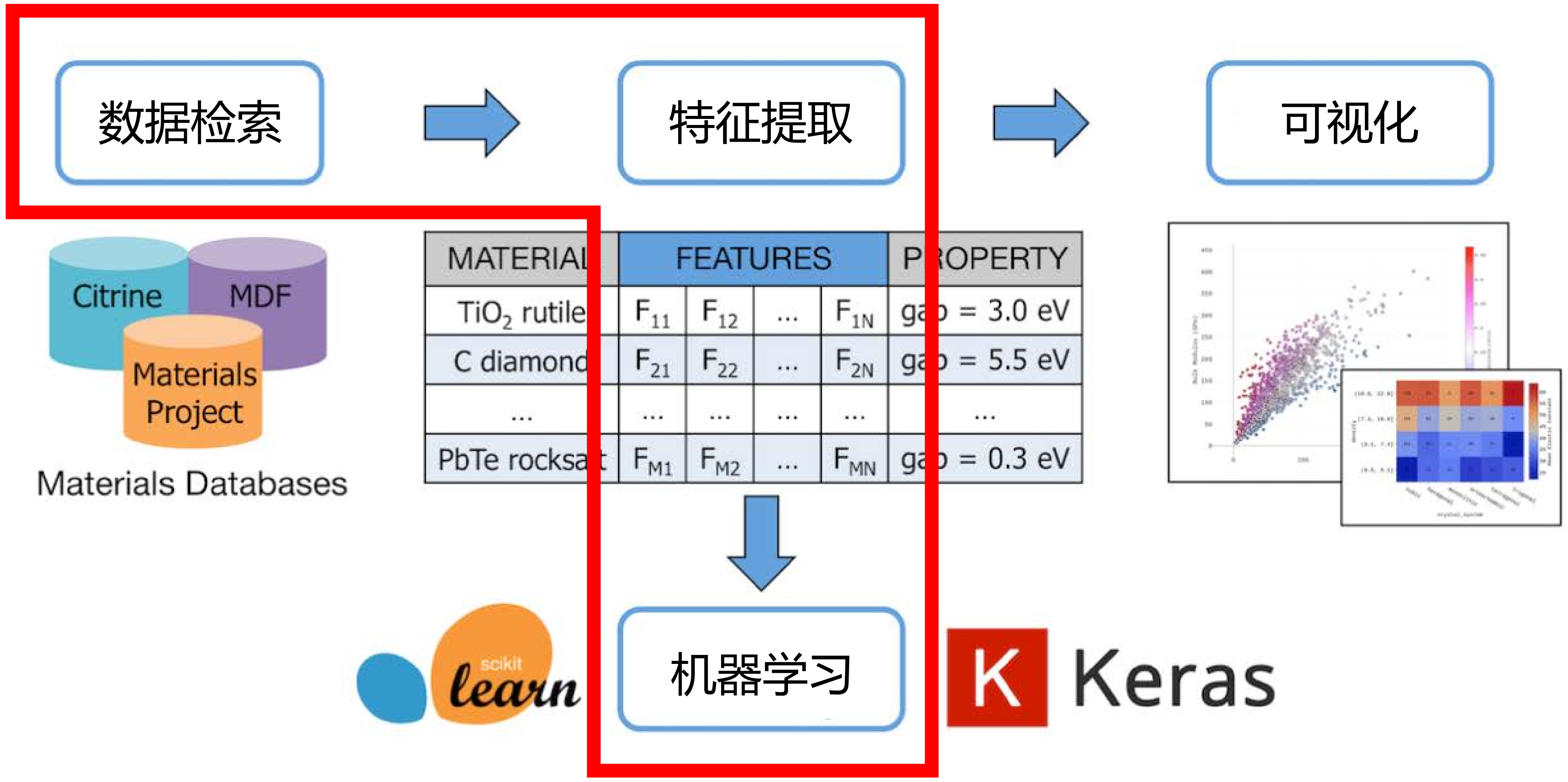
MATERIAL	FEATURES				PROPERTY
TiO_2 rutile	F_{11}	F_{12}	...	F_{1N}	gap = 3.0 eV
C diamond	F_{21}	F_{22}	...	F_{2N}	gap = 5.5 eV
...	...	...	...	...	...
PbTe rocksalt	F_{M1}	F_{M2}	...	F_{MN}	gap = 0.3 eV

Ward, L., Jain, A., *et al. Comput. Mater. Sci.* **2018**, 152, 60-69.

matminer机器学习接口

matminer

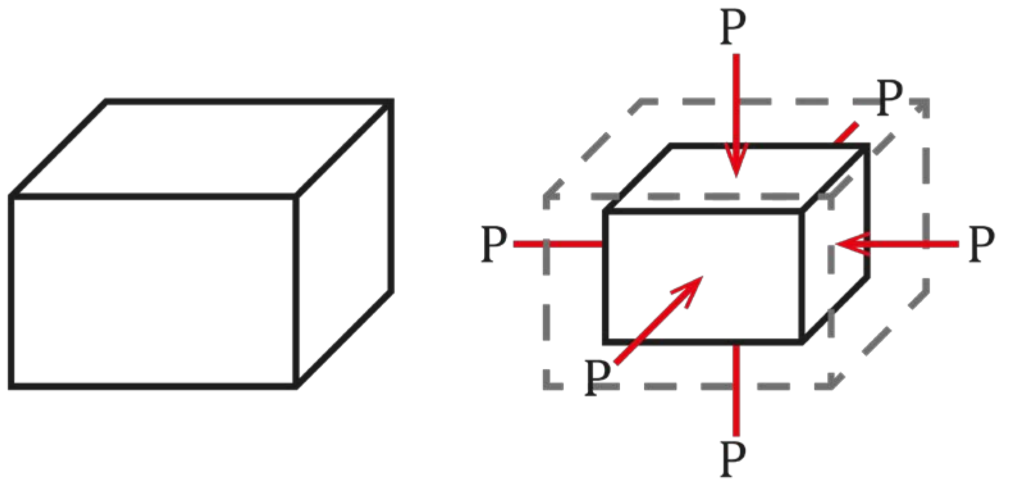
https://hackingmaterials.lbl.gov/matminer



MATERIAL	FEATURES				P OPERTY
TiO_2 rutile	F_{11}	F_{12}	…	F_{1N}	ga = 3.0 eV
C diamond	F_{21}	F_{22}	…	F_{2N}	ga = 5.5 eV
…	…	…	…	…	…
PbTe rocksa t	F_{M1}	F_{M2}	…	F_{MN}	ga = 0.3 eV

Ward, L., Jain, A., *et al. Comput. Mater. Sci.* **2018**, 152, 60-69.

体积模量



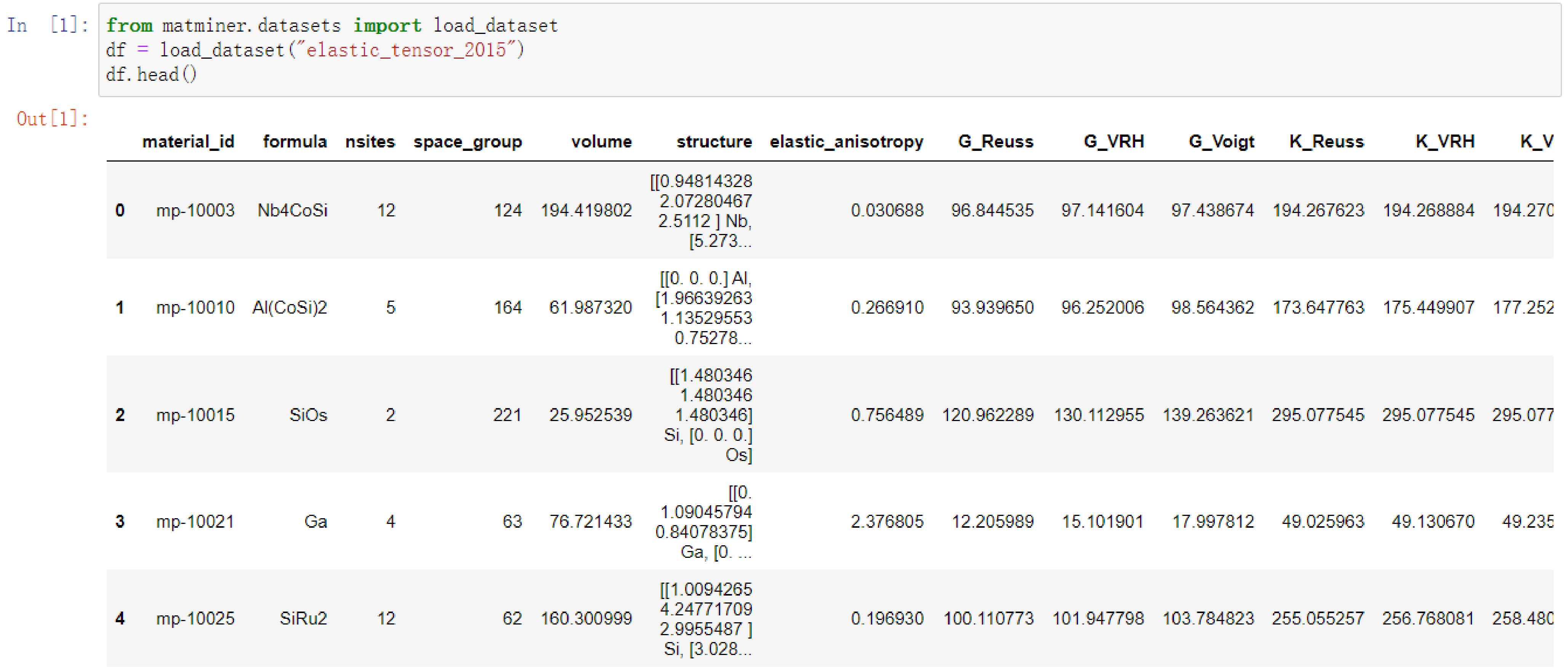
$$K = -V\frac{\partial p}{\partial V}$$

体积模量 （K）也称为不可压缩量，是材料对于表面四周压强产生形变程度的度量。

定义：产生单位相对体积收缩所需的压强，在SI单位制中的基本单位是帕斯卡。

预测体积模量

导入数据库 elastic_tensor_2015



In [1]:
```python
from matminer.datasets import load_dataset
df = load_dataset("elastic_tensor_2015")
df.head()
```

Out[1]:

	material_id	formula	nsites	space_group	volume	structure	elastic_anisotropy	G_Reuss	G_VRH	G_Voigt	K_Reuss	K_VRH	K_V
0	mp-10003	Nb4CoSi	12	124	194.419802	[[0.94814328 2.07280467 2.5112] Nb, [5.273...	0.030688	96.844535	97.141604	97.438674	194.267623	194.268884	194.270
1	mp-10010	Al(CoSi)2	5	164	61.987320	[[0. 0. 0.] Al, [1.96639263 1.13529553 0.75278...	0.266910	93.939650	96.252006	98.564362	173.647763	175.449907	177.252
2	mp-10015	SiOs	2	221	25.952539	[[1.480346 1.480346 1.480346] Si, [0. 0. 0.] Os]	0.756489	120.962289	130.112955	139.263621	295.077545	295.077545	295.077
3	mp-10021	Ga	4	63	76.721433	[[0. 1.09045794 0.84078375] Ga, [0. ...	2.376805	12.205989	15.101901	17.997812	49.025963	49.130670	49.235
4	mp-10025	SiRu2	12	62	160.300999	[[1.0094265 4.24771709 2.9955487] Si, [3.028...	0.196930	100.110773	101.947798	103.784823	255.055257	256.768081	258.480

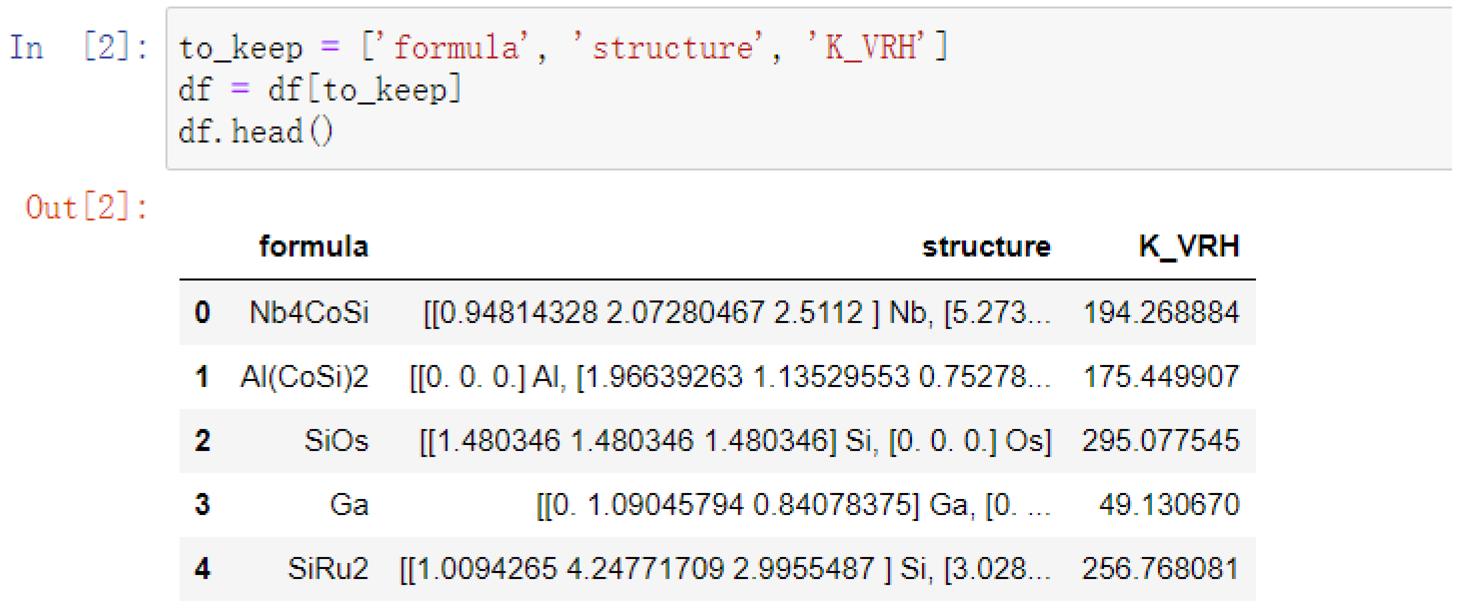
查看数据库信息

包含了1181个使用DFT-PBE计算的包含弹性性质的结构

列	描述
G_Reuss	多晶材料的剪切模量下界
G_VRH	G_Reuss与G_Voigt的平均值
G_Voigt	多晶材料的剪切模量上界
K_Reuss	多晶材料的体积模量下界
K_VRH	K_Reuss与K_Voigt的平均值
K_Voigt	多晶材料的体积模量上界
cif	可选：结构的描述字符串
compliance_tensor	描述弹性行为的张量
elastic_anisotropy	材料弹性方向依赖性的度量，度量总是 >= 0
elastic_tensor	描述对应于 IEEE 方向的弹性行为的张量，对称于晶体结构
elastic_tensor_original	描述弹性行为的张量，非对称的，对应于 POSCAR 常规标准单元方向

列	描述
formula	材料的化学组成
kpoint_density	可选：计算中的采样参数
material_id	材料的Materials Project ID
nsites	计算单胞的原子数
poisson_ratio	描述对负载的横向响应
poscar	可选：POSCAR数据
space_group	材料晶体结构的空间群
structure	pandas 系列定义了材料的结构
volume	以立方埃为单位的晶胞体积，对于超晶胞计算，这个量是指整个超晶胞的体积。

整理筛选信息



In [2]:
```
to_keep = ['formula', 'structure', 'K_VRH']
df = df[to_keep]
df.head()
```

Out[2]:

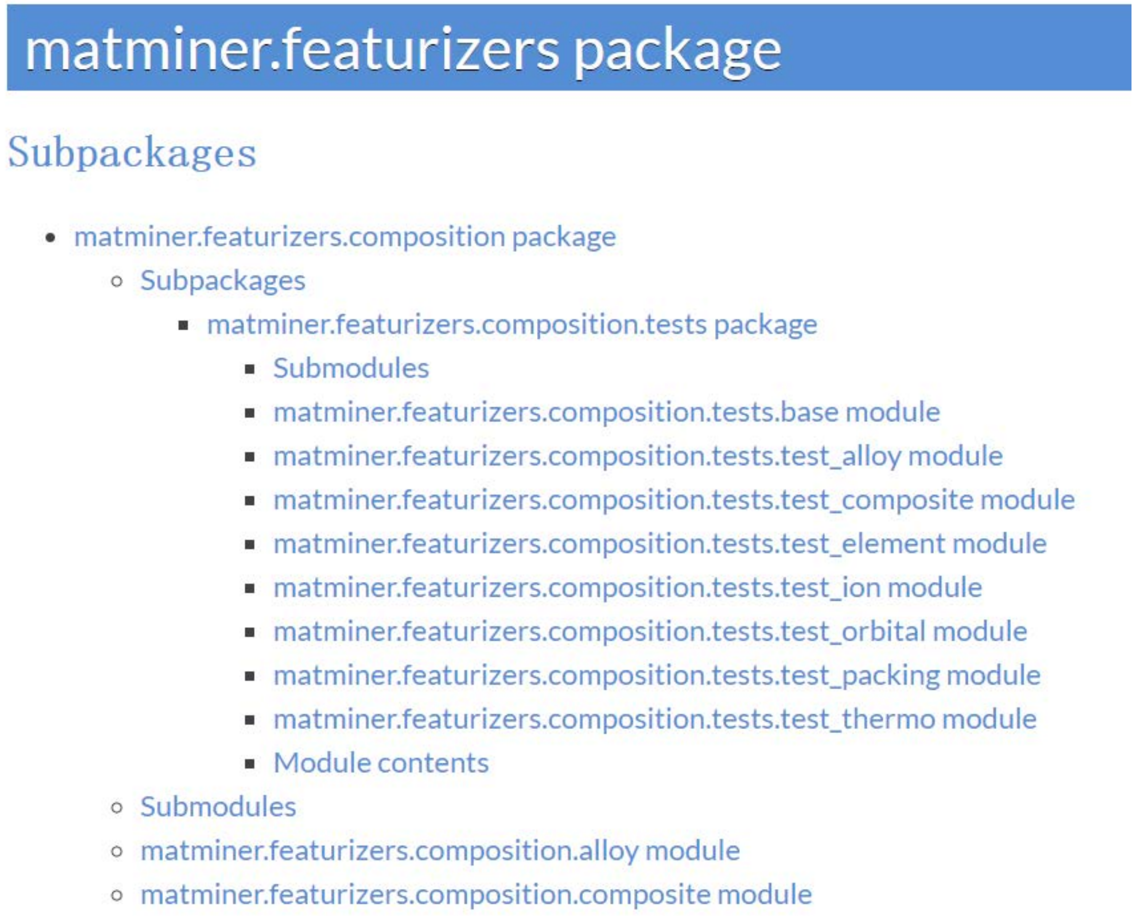
	formula	structure	K_VRH
0	Nb4CoSi	[[0.94814328 2.07280467 2.5112] Nb, [5.273...	194.268884
1	Al(CoSi)2	[[0. 0. 0.] Al, [1.96639263 1.13529553 0.75278...	175.449907
2	SiOs	[[1.480346 1.480346 1.480346] Si, [0. 0. 0.] Os]	295.077545
3	Ga	[[0. 1.09045794 0.84078375] Ga, [0. ...	49.130670
4	SiRu2	[[1.0094265 4.24771709 2.9955487] Si, [3.028...	256.768081

用于添加相关描述符

用于回归

添加描述符

featurizers模块　　from matminer.featurizers.conversions import StrToComposition



matminer.featurizers package

Subpackages

- matminer.featurizers.composition package
  - Subpackages
    - matminer.featurizers.composition.tests package
      - Submodules
      - matminer.featurizers.composition.tests.base module
      - matminer.featurizers.composition.tests.test_alloy module
      - matminer.featurizers.composition.tests.test_composite module
      - matminer.featurizers.composition.tests.test_element module
      - matminer.featurizers.composition.tests.test_ion module
      - matminer.featurizers.composition.tests.test_orbital module
      - matminer.featurizers.composition.tests.test_packing module
      - matminer.featurizers.composition.tests.test_thermo module
      - Module contents
  - Submodules
  - matminer.featurizers.composition.alloy module
  - matminer.featurizers.composition.composite module

用于向dataframe中添加描述符，例如：

元素种类

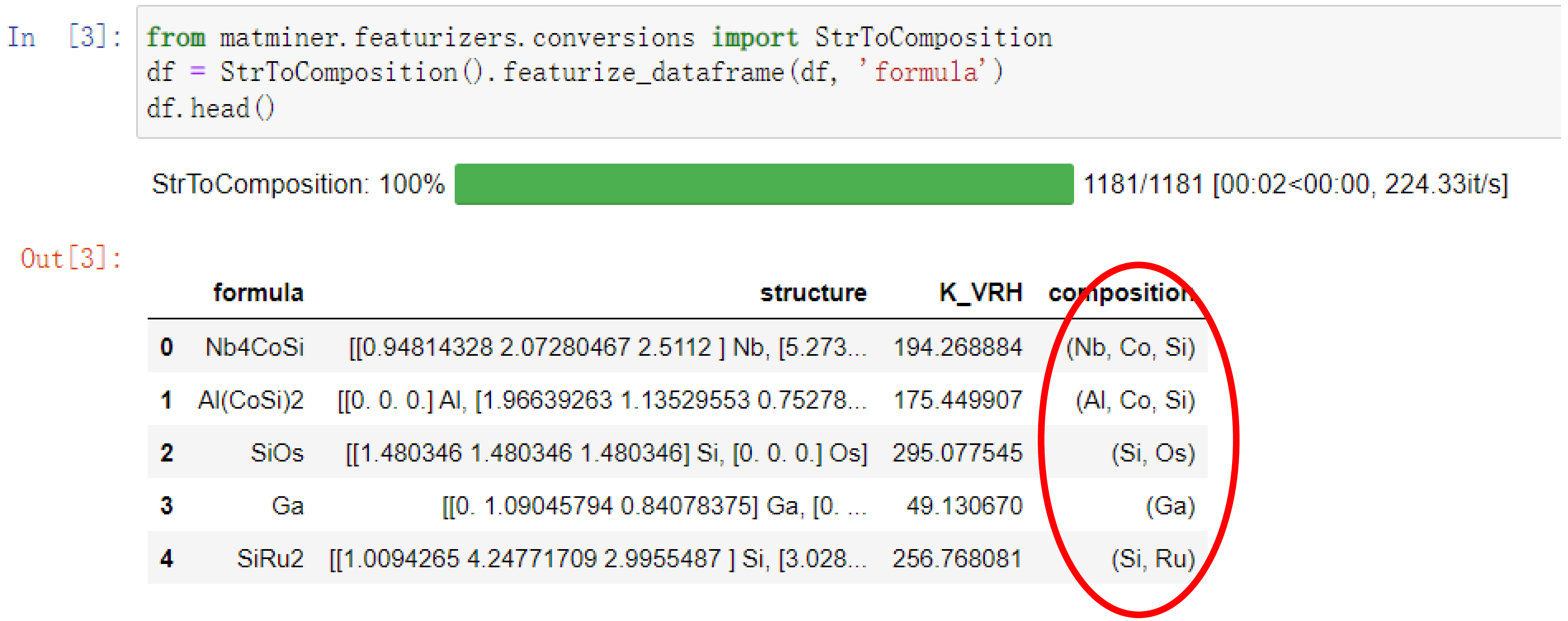
氧化态

态密度（DOS）

……

添加描述符

添加元素信息



In [3]:
```
from matminer.featurizers.conversions import StrToComposition
df = StrToComposition().featurize_dataframe(df, 'formula')
df.head()
```

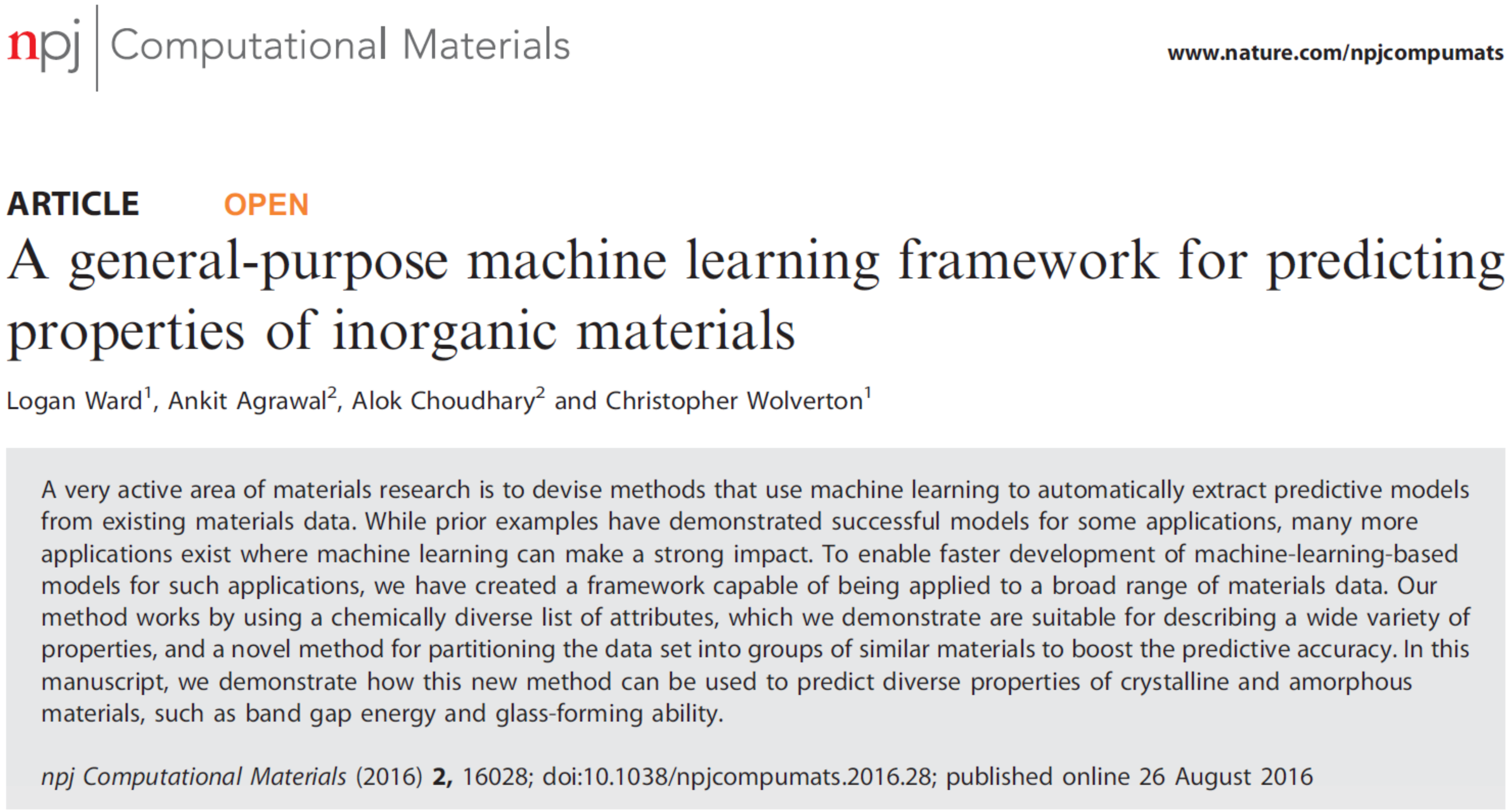
```
StrToComposition: 100% 1181/1181 [00:02<00:00, 224.33it/s]
```

Out[3]:

	formula	structure	K_VRH	composition
0	Nb4CoSi	[[0.94814328 2.07280467 2.5112] Nb, [5.273...	194.268884	(Nb, Co, Si)
1	Al(CoSi)2	[[0. 0. 0.] Al, [1.96639263 1.13529553 0.75278...	175.449907	(Al, Co, Si)
2	SiOs	[[1.480346 1.480346 1.480346] Si, [0. 0. 0.] Os]	295.077545	(Si, Os)
3	Ga	[[0. 1.09045794 0.84078375] Ga, [0. ...	49.130670	(Ga)
4	SiRu2	[[1.0094265 4.24771709 2.9955487] Si, [3.028...	256.768081	(Si, Ru)

Magpie属性

https://hachmannlab.github.io/chemml/chemml.chem.magpie_python.html



npj Computational Materials www.nature.com/npjcompumats

ARTICLE OPEN

A general-purpose machine learning framework for predicting properties of inorganic materials

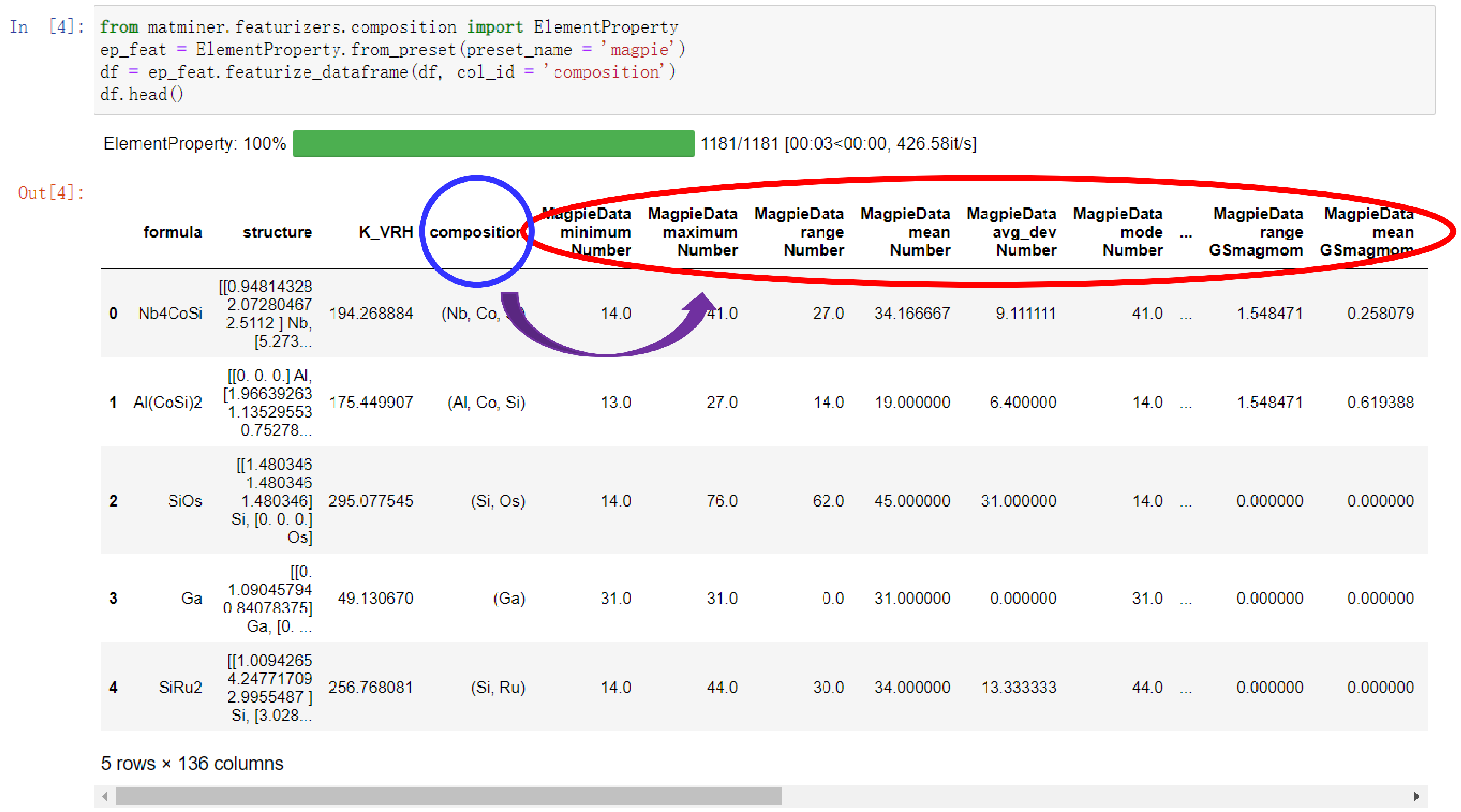
Logan Ward[1], Ankit Agrawal[2], Alok Choudhary[2] and Christopher Wolverton[1]

A very active area of materials research is to devise methods that use machine learning to automatically extract predictive models from existing materials data. While prior examples have demonstrated successful models for some applications, many more applications exist where machine learning can make a strong impact. To enable faster development of machine-learning-based models for such applications, we have created a framework capable of being applied to a broad range of materials data. Our method works by using a chemically diverse list of attributes, which we demonstrate are suitable for describing a wide variety of properties, and a novel method for partitioning the data set into groups of similar materials to boost the predictive accuracy. In this manuscript, we demonstrate how this new method can be used to predict diverse properties of crystalline and amorphous materials, such as band gap energy and glass-forming ability.

npj Computational Materials (2016) **2,** 16028; doi:10.1038/npjcompumats.2016.28; published online 26 August 2016

添加描述符

添加基础特征



In [4]:
```
from matminer.featurizers.composition import ElementProperty
ep_feat = ElementProperty.from_preset(preset_name = 'magpie')
df = ep_feat.featurize_dataframe(df, col_id = 'composition')
df.head()
```

```
ElementProperty: 100% 🟩 1181/1181 [00:03<00:00, 426.58it/s]
```

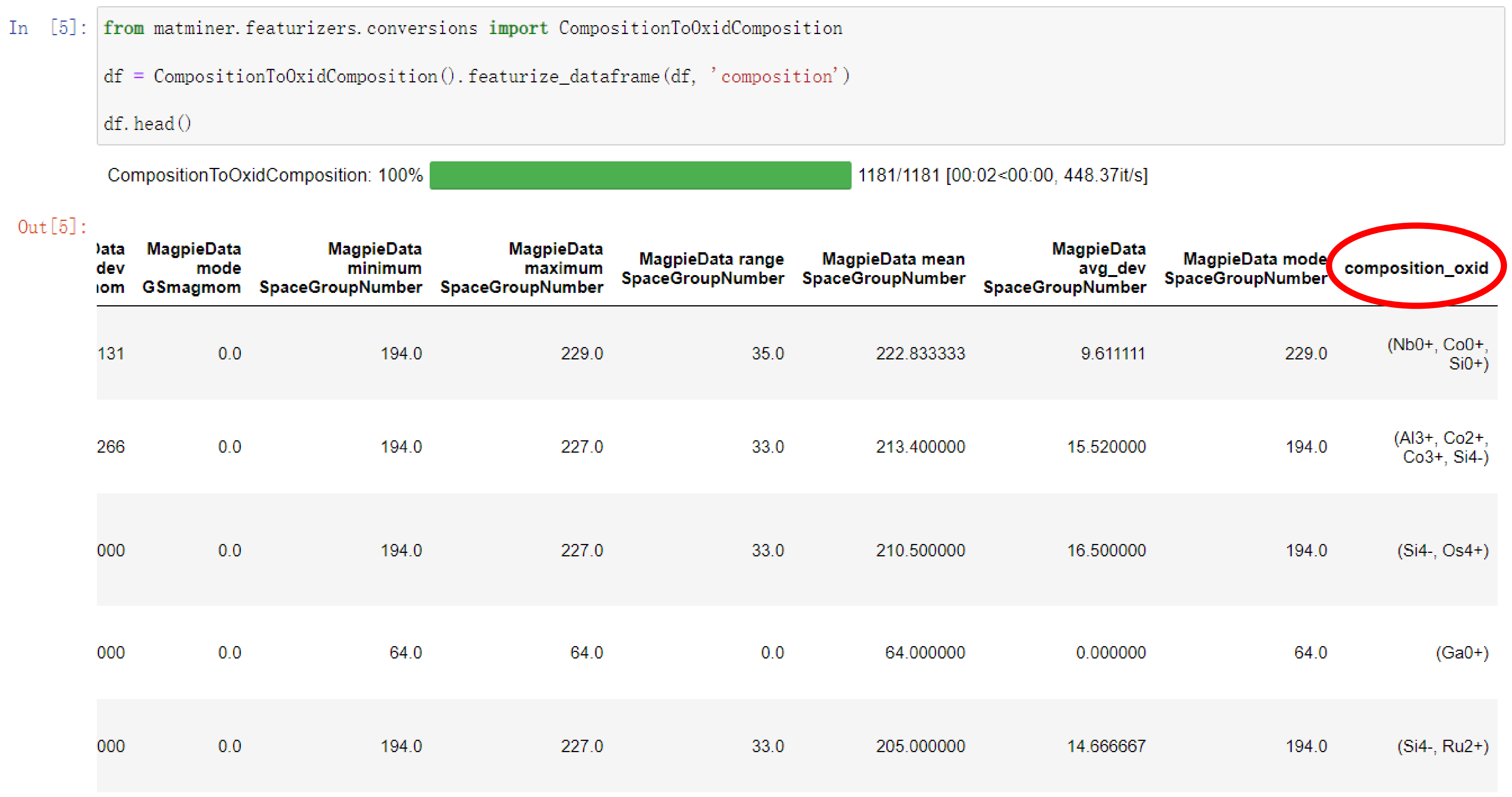
Out[4]:

	formula	structure	K_VRH	composition	MagpieData minimum Number	MagpieData maximum Number	MagpieData range Number	MagpieData mean Number	MagpieData avg_dev Number	MagpieData mode Number	...	MagpieData range GSmagmom	MagpieData mean GSmagmom
0	Nb4CoSi	[[0.94814328 2.07280467 2.5112] Nb, [5.273...	194.268884	(Nb, Co, Si)	14.0	41.0	27.0	34.166667	9.111111	41.0	...	1.548471	0.258079
1	Al(CoSi)2	[[0. 0. 0.] Al, [1.96639263 1.13529553 0.75278...	175.449907	(Al, Co, Si)	13.0	27.0	14.0	19.000000	6.400000	14.0	...	1.548471	0.619388
2	SiOs	[[1.480346 1.480346 1.480346] Si, [0. 0. 0.] Os]	295.077545	(Si, Os)	14.0	76.0	62.0	45.000000	31.000000	14.0	...	0.000000	0.000000
3	Ga	[[0. 1.09045794 0.84078375] Ga, [0. ...	49.130670	(Ga)	31.0	31.0	0.0	31.000000	0.000000	31.0	...	0.000000	0.000000
4	SiRu2	[[1.0094265 4.24771709 2.9955487] Si, [3.028...	256.768081	(Si, Ru)	14.0	44.0	30.0	34.000000	13.333333	44.0	...	0.000000	0.000000

5 rows × 136 columns

添加描述符

添加组分特征



```
In [5]: from matminer.featurizers.conversions import CompositionToOxidComposition

        df = CompositionToOxidComposition().featurize_dataframe(df, 'composition')

        df.head()
```

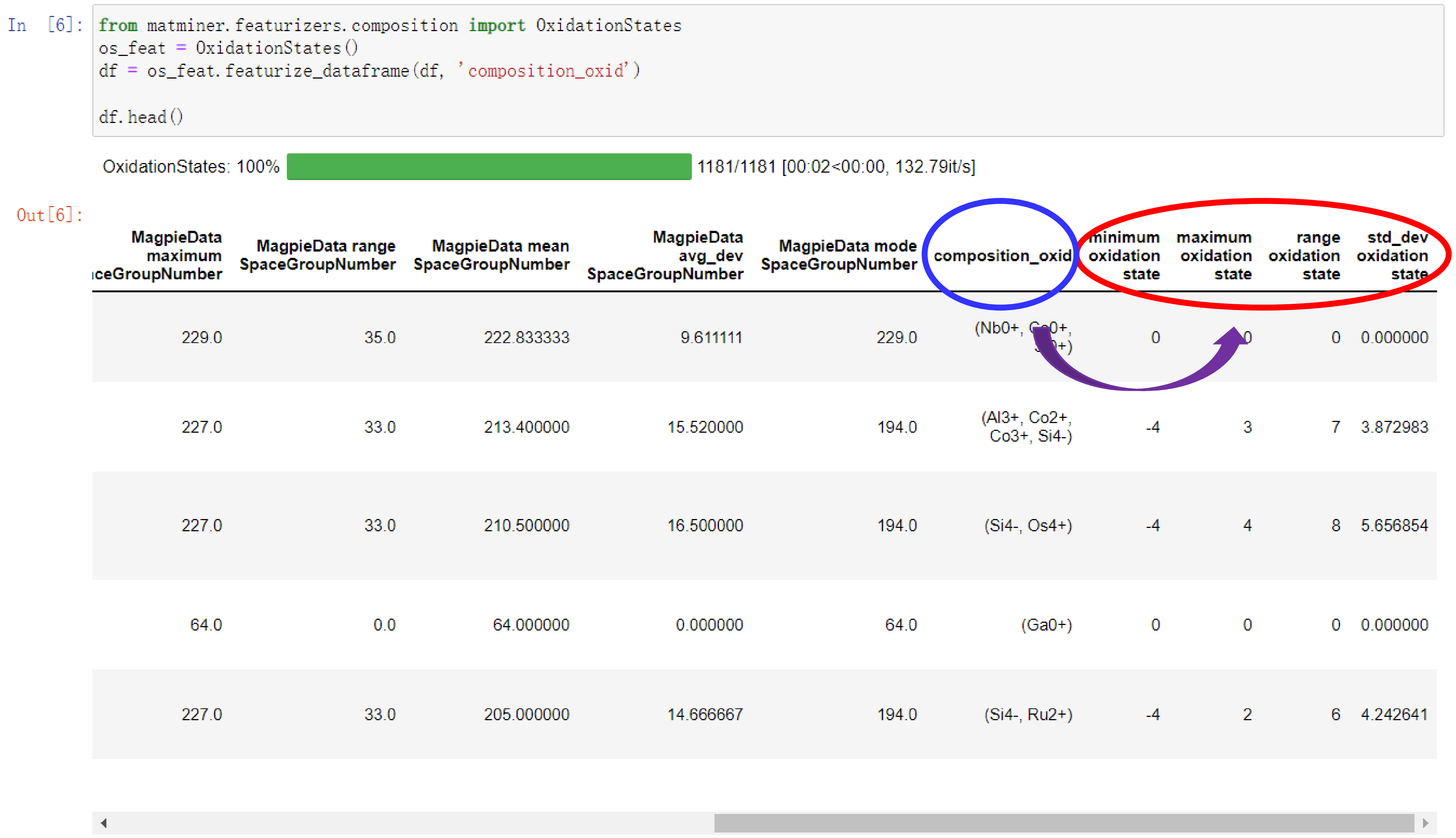
```
CompositionToOxidComposition: 100% 1181/1181 [00:02<00:00, 448.37it/s]
```

Out[5]:

ata dev om	MagpieData mode GSmagmom	MagpieData minimum SpaceGroupNumber	MagpieData maximum SpaceGroupNumber	MagpieData range SpaceGroupNumber	MagpieData mean SpaceGroupNumber	MagpieData avg_dev SpaceGroupNumber	MagpieData mode SpaceGroupNumber	composition_oxid
131	0.0	194.0	229.0	35.0	222.833333	9.611111	229.0	(Nb0+, Co0+, Si0+)
266	0.0	194.0	227.0	33.0	213.400000	15.520000	194.0	(Al3+, Co2+, Co3+, Si4-)
000	0.0	194.0	227.0	33.0	210.500000	16.500000	194.0	(Si4-, Os4+)
000	0.0	64.0	64.0	0.0	64.000000	0.000000	64.0	(Ga0+)
000	0.0	194.0	227.0	33.0	205.000000	14.666667	194.0	(Si4-, Ru2+)

添加描述符

添加组分特征



```
In [6]: from matminer.featurizers.composition import OxidationStates
        os_feat = OxidationStates()
        df = os_feat.featurize_dataframe(df, 'composition_oxid')

        df.head()
```

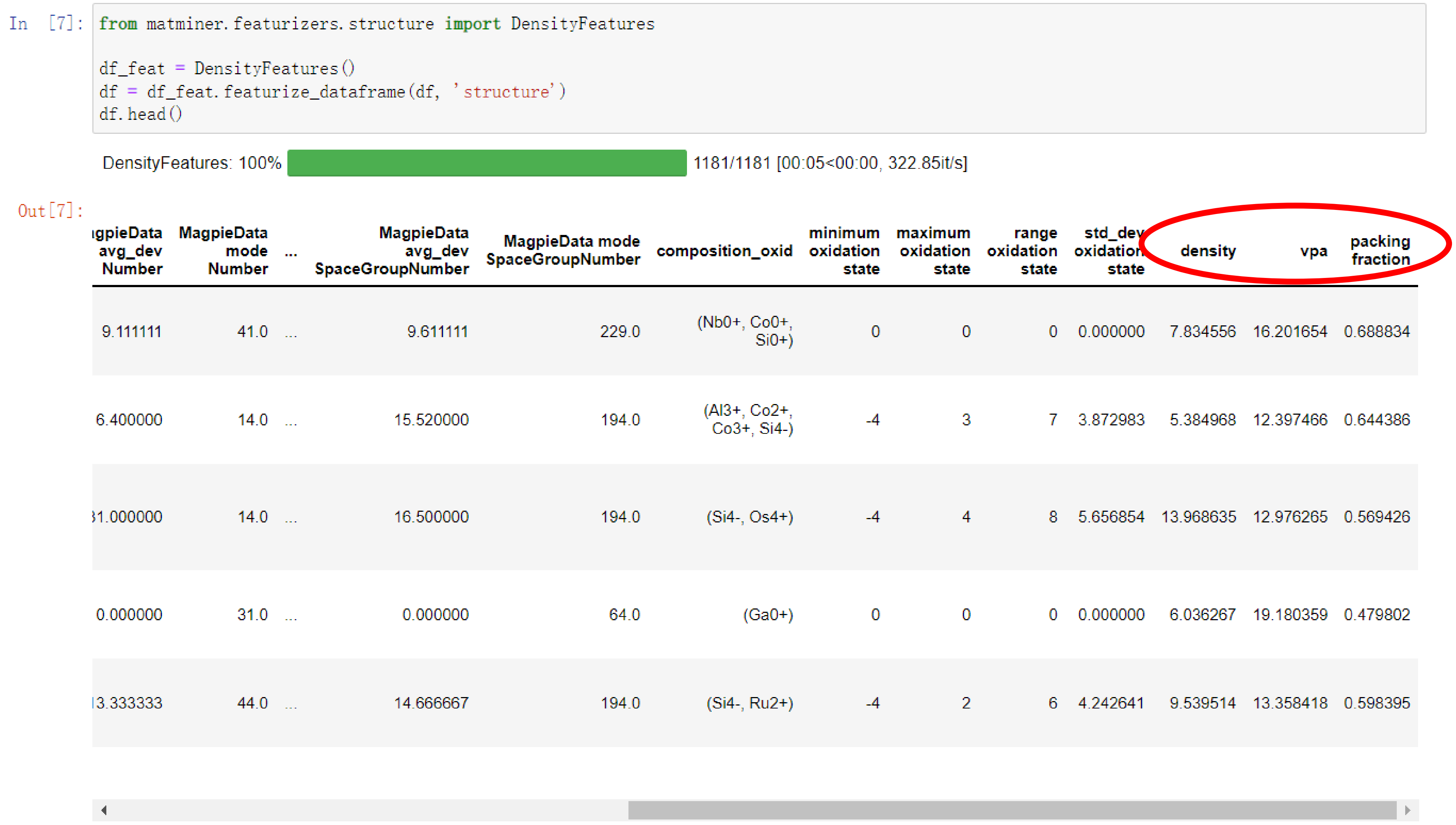
```
OxidationStates: 100%  1181/1181 [00:02<00:00, 132.79it/s]
```

Out[6]:

MagpieData maximum ceGroupNumber	MagpieData range SpaceGroupNumber	MagpieData mean SpaceGroupNumber	MagpieData avg_dev SpaceGroupNumber	MagpieData mode SpaceGroupNumber	composition_oxid	minimum oxidation state	maximum oxidation state	range oxidation state	std_dev oxidation state
229.0	35.0	222.833333	9.611111	229.0	(Nb0+, Co0+, Si0+)	0	0	0	0.000000
227.0	33.0	213.400000	15.520000	194.0	(Al3+, Co2+, Co3+, Si4-)	-4	3	7	3.872983
227.0	33.0	210.500000	16.500000	194.0	(Si4-, Os4+)	-4	4	8	5.656854
64.0	0.0	64.000000	0.000000	64.0	(Ga0+)	0	0	0	0.000000
227.0	33.0	205.000000	14.666667	194.0	(Si4-, Ru2+)	-4	2	6	4.242641

添加描述符

添加结构特征



In [7]:
```
from matminer.featurizers.structure import DensityFeatures

df_feat = DensityFeatures()
df = df_feat.featurize_dataframe(df, 'structure')
df.head()
```

```
DensityFeatures: 100% 🟩 1181/1181 [00:05<00:00, 322.85it/s]
```

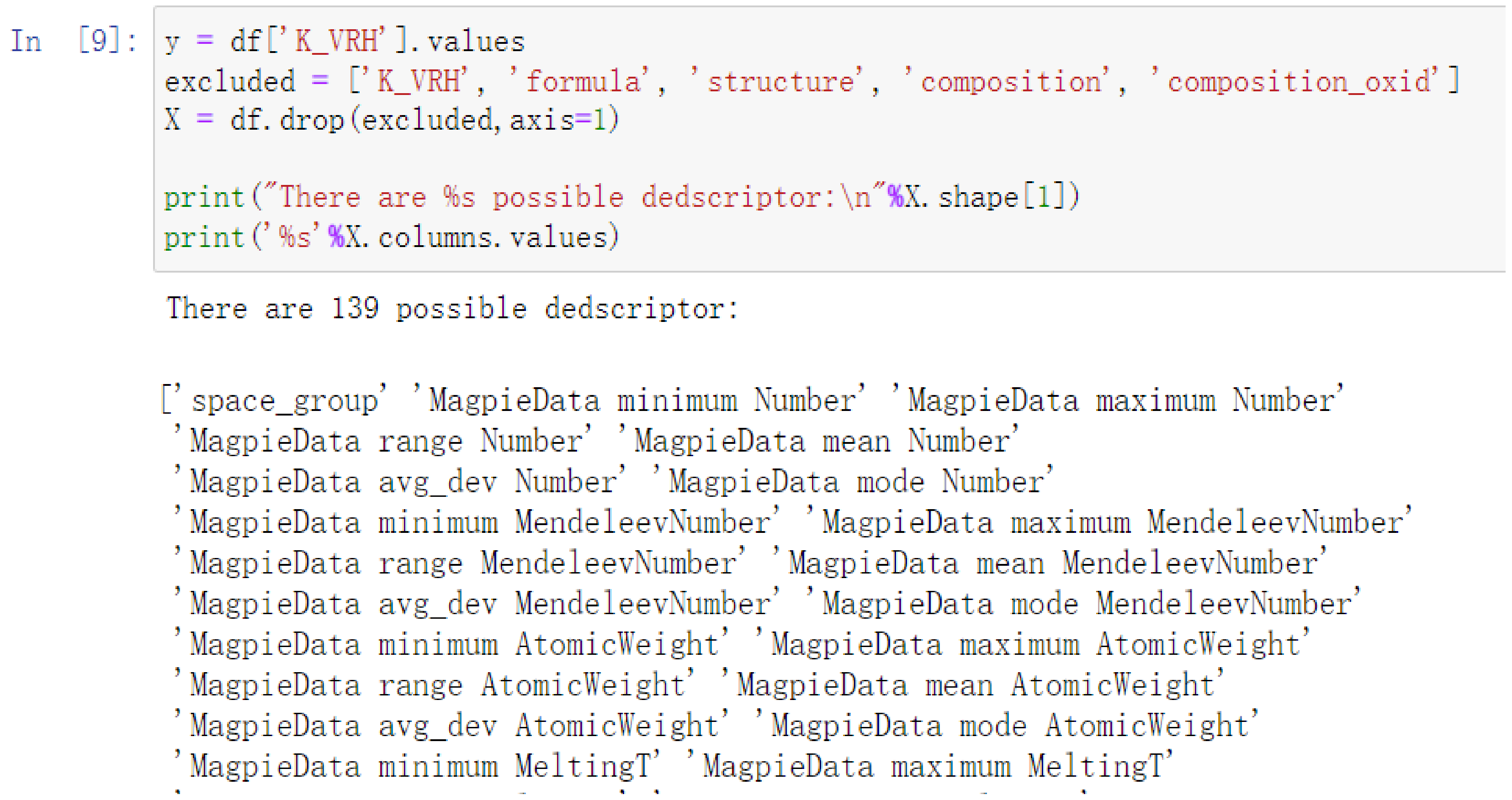
Out[7]:

gpieData avg_dev Number	MagpieData mode Number	...	MagpieData avg_dev SpaceGroupNumber	MagpieData mode SpaceGroupNumber	composition_oxid	minimum oxidation state	maximum oxidation state	range oxidation state	std_dev oxidation state	density	vpa	packing fraction
9.111111	41.0	...	9.611111	229.0	(Nb0+, Co0+, Si0+)	0	0	0	0.000000	7.834556	16.201654	0.688834
6.400000	14.0	...	15.520000	194.0	(Al3+, Co2+, Co3+, Si4-)	-4	3	7	3.872983	5.384968	12.397466	0.644386
1.000000	14.0	...	16.500000	194.0	(Si4-, Os4+)	-4	4	8	5.656854	13.968635	12.976265	0.569426
0.000000	31.0	...	0.000000	64.0	(Ga0+)	0	0	0	0.000000	6.036267	19.180359	0.479802
3.333333	44.0	...	14.666667	194.0	(Si4-, Ru2+)	-4	2	6	4.242641	9.539514	13.358418	0.598395

定义输入输出

输出：体积模量K_VRH，也可使用G_VRH，elastic_anisotropy，possion_ratio作为输出

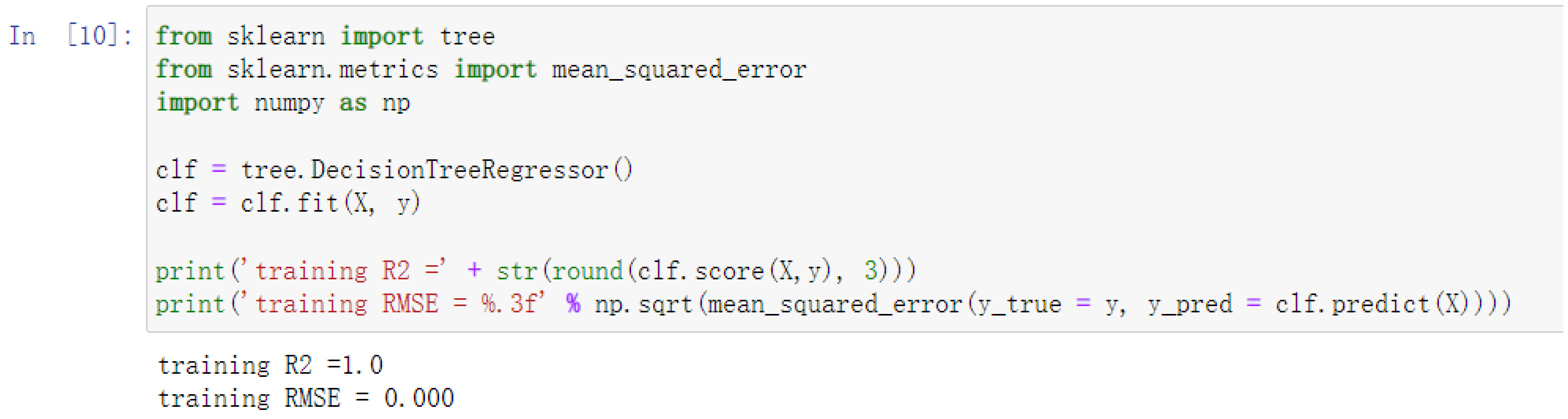
输入：除了输出数据、非数字数据，都作为输入



```
In  [9]: y = df['K_VRH'].values
         excluded = ['K_VRH', 'formula', 'structure', 'composition', 'composition_oxid']
         X = df.drop(excluded,axis=1)

         print("There are %s possible dedscriptor:\n"%X.shape[1])
         print('%s'%X.columns.values)
```

```
There are 139 possible dedscriptor:

['space_group' 'MagpieData minimum Number' 'MagpieData maximum Number'
 'MagpieData range Number' 'MagpieData mean Number'
 'MagpieData avg_dev Number' 'MagpieData mode Number'
 'MagpieData minimum MendeleevNumber' 'MagpieData maximum MendeleevNumber'
 'MagpieData range MendeleevNumber' 'MagpieData mean MendeleevNumber'
 'MagpieData avg_dev MendeleevNumber' 'MagpieData mode MendeleevNumber'
 'MagpieData minimum AtomicWeight' 'MagpieData maximum AtomicWeight'
 'MagpieData range AtomicWeight' 'MagpieData mean AtomicWeight'
 'MagpieData avg_dev AtomicWeight' 'MagpieData mode AtomicWeight'
 'MagpieData minimum MeltingT' 'MagpieData maximum MeltingT'
```

决策树回归

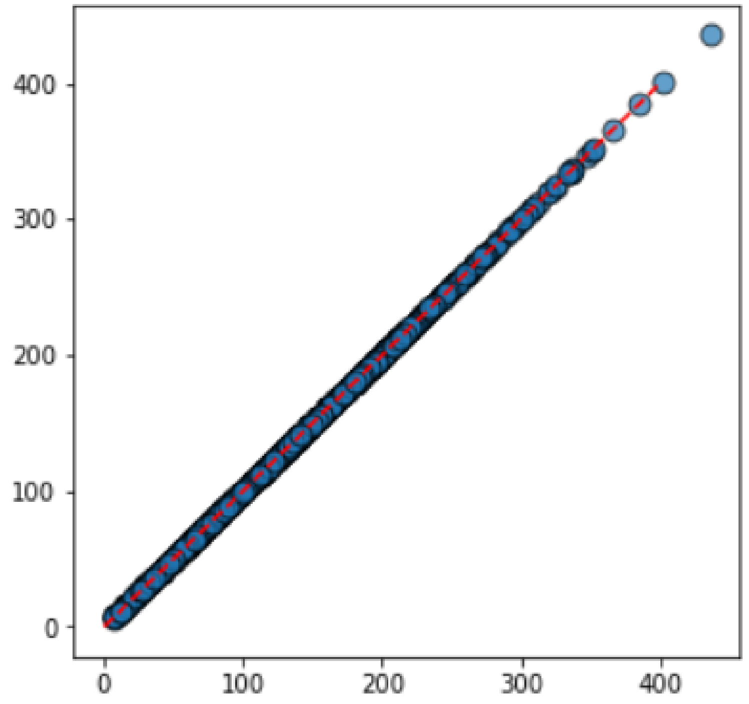


```
In  [10]: from sklearn import tree
          from sklearn.metrics import mean_squared_error
          import numpy as np

          clf = tree.DecisionTreeRegressor()
          clf = clf.fit(X, y)

          print('training R2 =' + str(round(clf.score(X,y), 3)))
          print('training RMSE = %.3f' % np.sqrt(mean_squared_error(y_true = y, y_pred = clf.predict(X))))
```

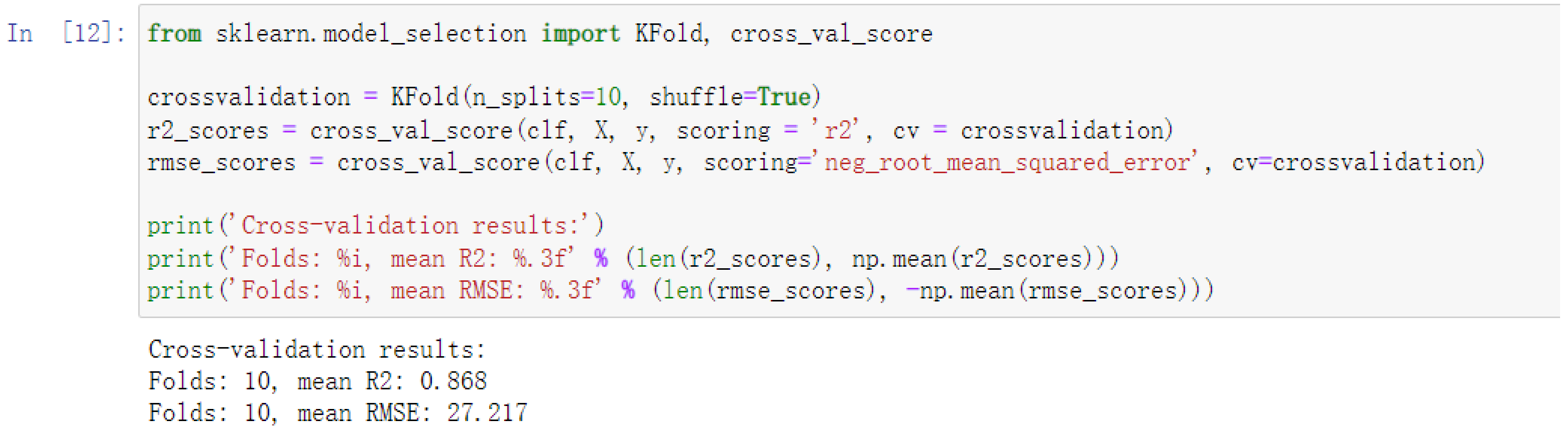
```
training R2 =1.0
training RMSE = 0.000
```

决策树回归

```
In  [11]: import matplotlib.pyplot as plt

          plt.figure(figsize=(5,5))
          plt.plot([0,400], [0, 400], 'r--')
          plt.scatter(y, clf.predict(X), s = 80, edgecolor = 'k', alpha = 0.7)

Out[11]: <matplotlib.collections.PathCollection at 0x267c6291fa0>
```



交叉验证

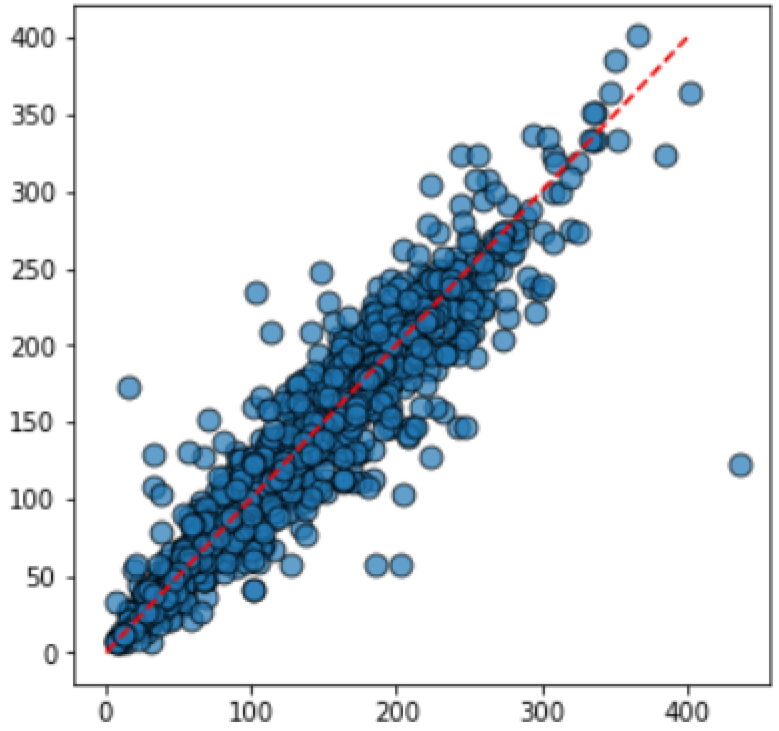


```
In  [12]: from sklearn.model_selection import KFold, cross_val_score

          crossvalidation = KFold(n_splits=10, shuffle=True)
          r2_scores = cross_val_score(clf, X, y, scoring = 'r2', cv = crossvalidation)
          rmse_scores = cross_val_score(clf, X, y, scoring='neg_root_mean_squared_error', cv=crossvalidation)

          print('Cross-validation results:')
          print('Folds: %i, mean R2: %.3f' % (len(r2_scores), np.mean(r2_scores)))
          print('Folds: %i, mean RMSE: %.3f' % (len(rmse_scores), -np.mean(rmse_scores)))
```

```
Cross-validation results:
Folds: 10, mean R2: 0.868
Folds: 10, mean RMSE: 27.217
```

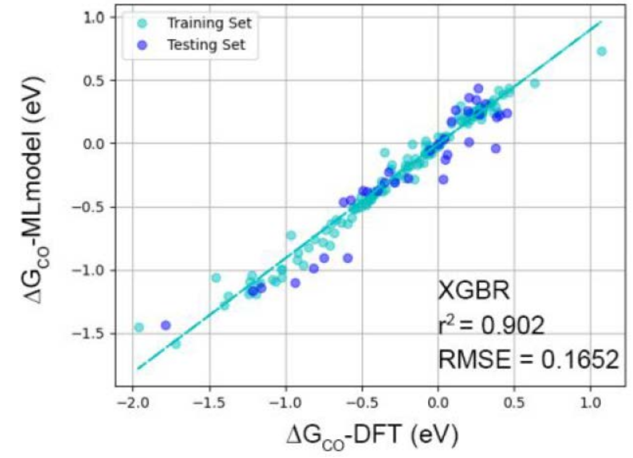
交叉验证

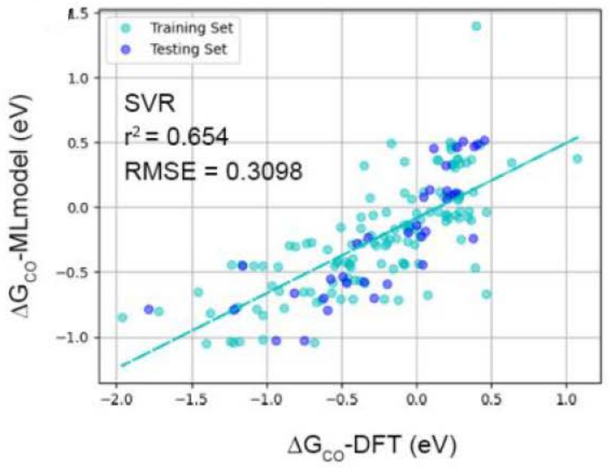
```
In  [14]: plt.figure(figsize=(5,5))
          plt.plot([0,400],[0,400],'r--')
          plt.scatter(y, y_cv, s = 80, c = None, edgecolor = 'k', alpha = 0.7)

Out[14]: <matplotlib.collections.PathCollection at 0x267bd108430>
```

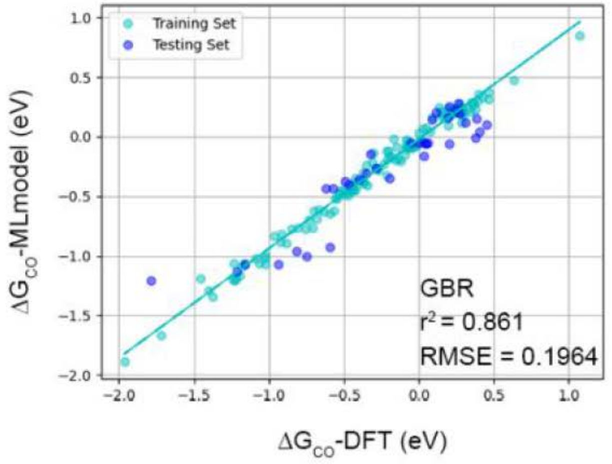


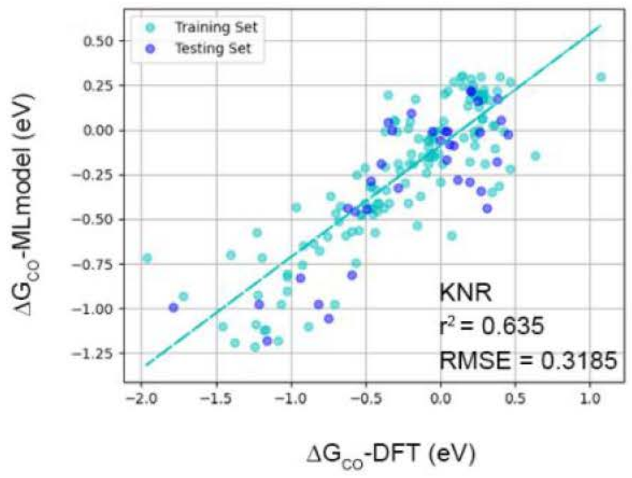
模型更改

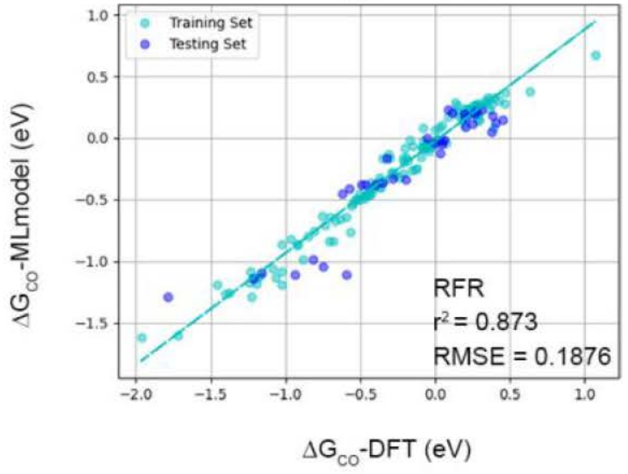
不同的模型可能对特定的体系表现会有很大的差别

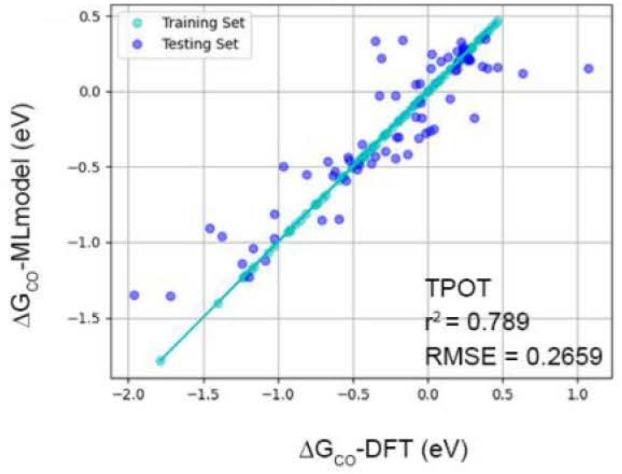






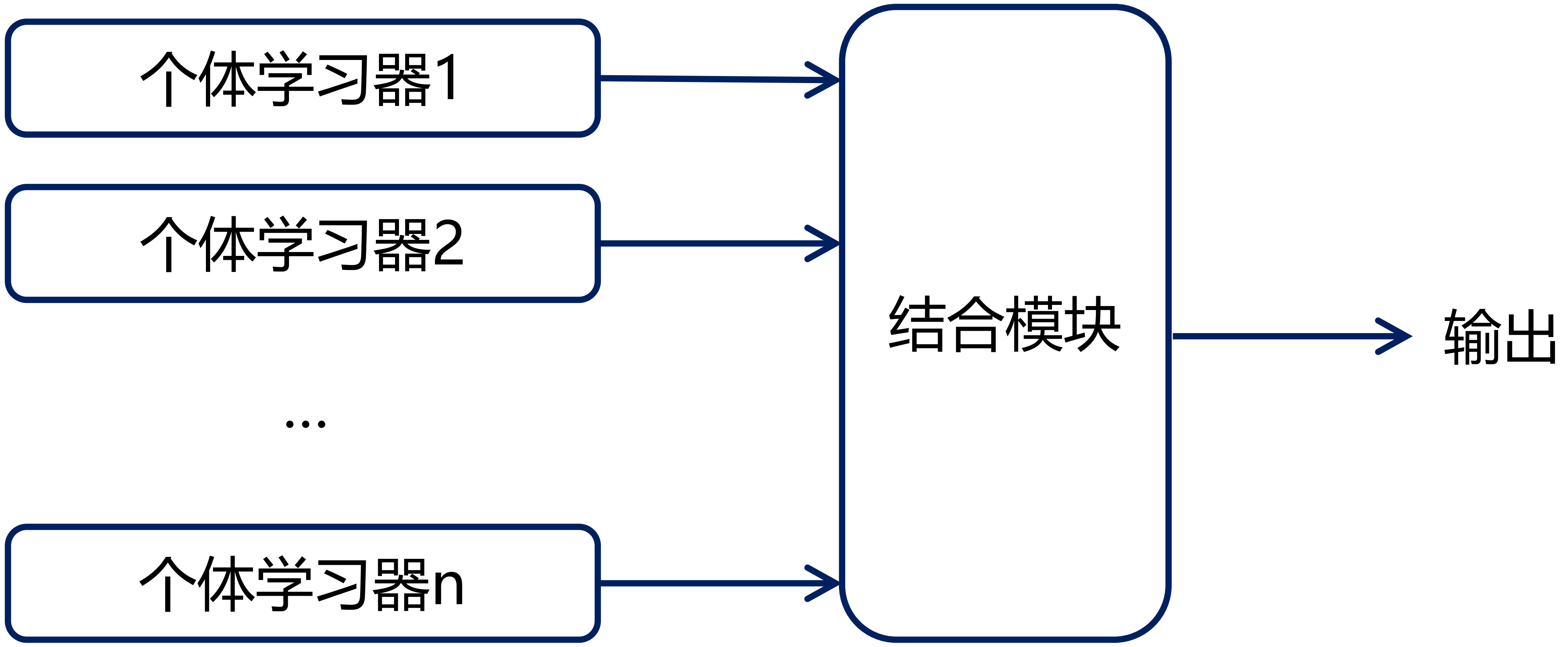






Chen, A., Zhou, Z., *et al. J. Phys. Chem. C* **2020**, 124, 22471-22478.

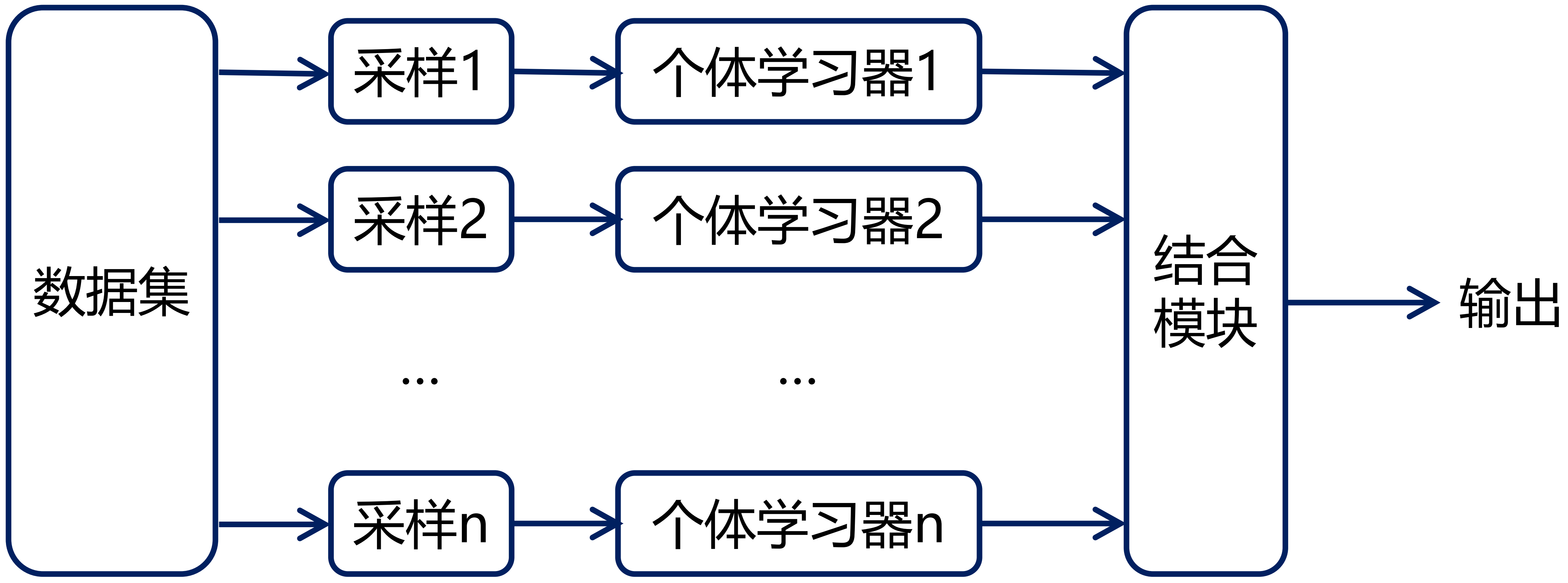
集成学习



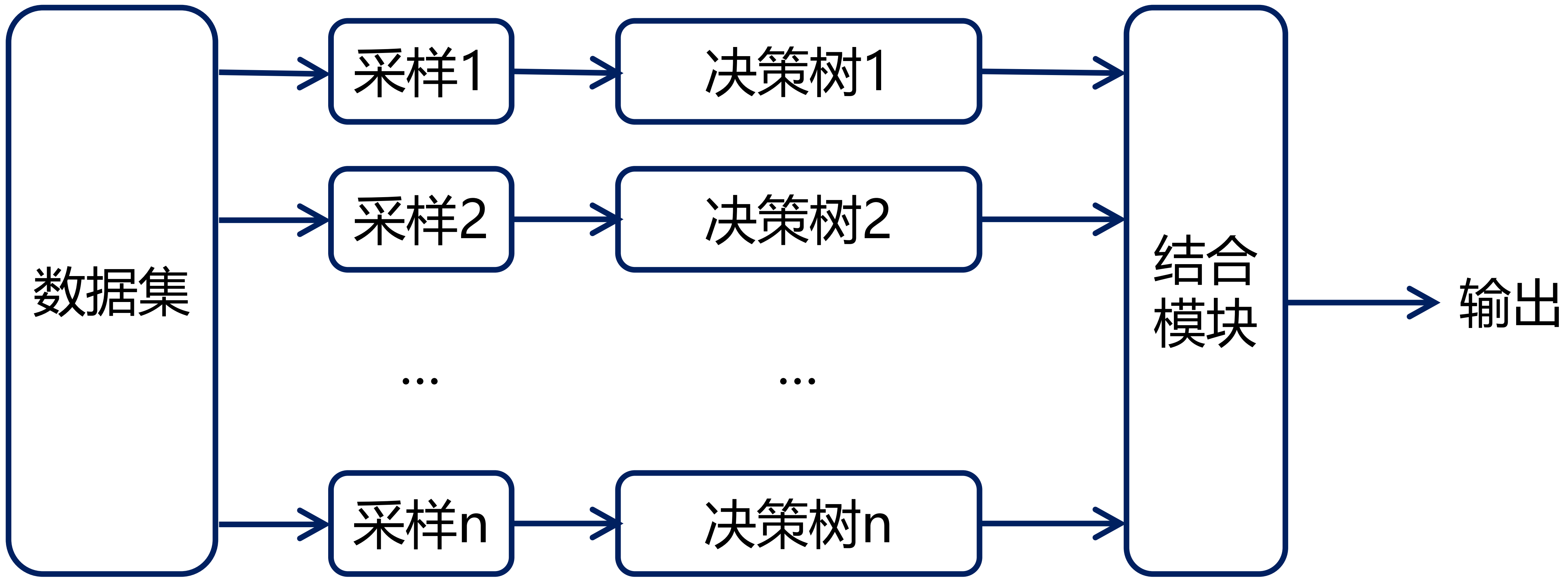
串行：Boosting

并行：Bagging

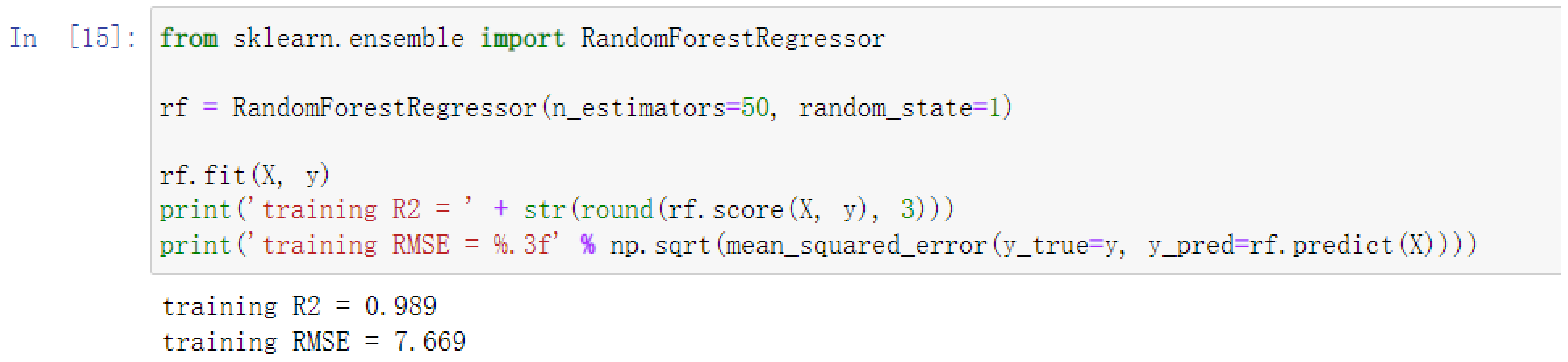
Bagging



随机森林算法



随机森林模型

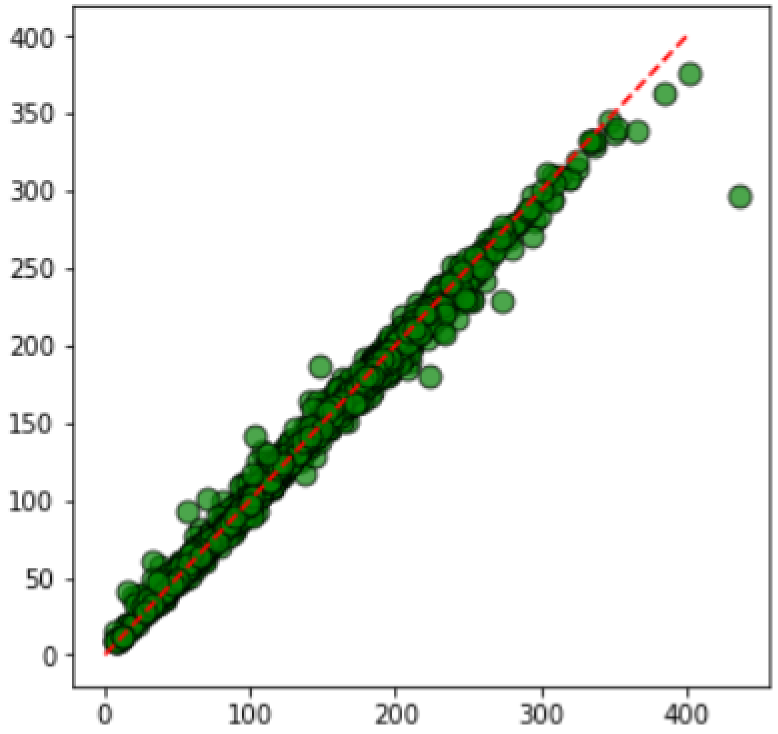


```
In [15]: from sklearn.ensemble import RandomForestRegressor

         rf = RandomForestRegressor(n_estimators=50, random_state=1)

         rf.fit(X, y)
         print('training R2 = ' + str(round(rf.score(X, y), 3)))
         print('training RMSE = %.3f' % np.sqrt(mean_squared_error(y_true=y, y_pred=rf.predict(X))))
```

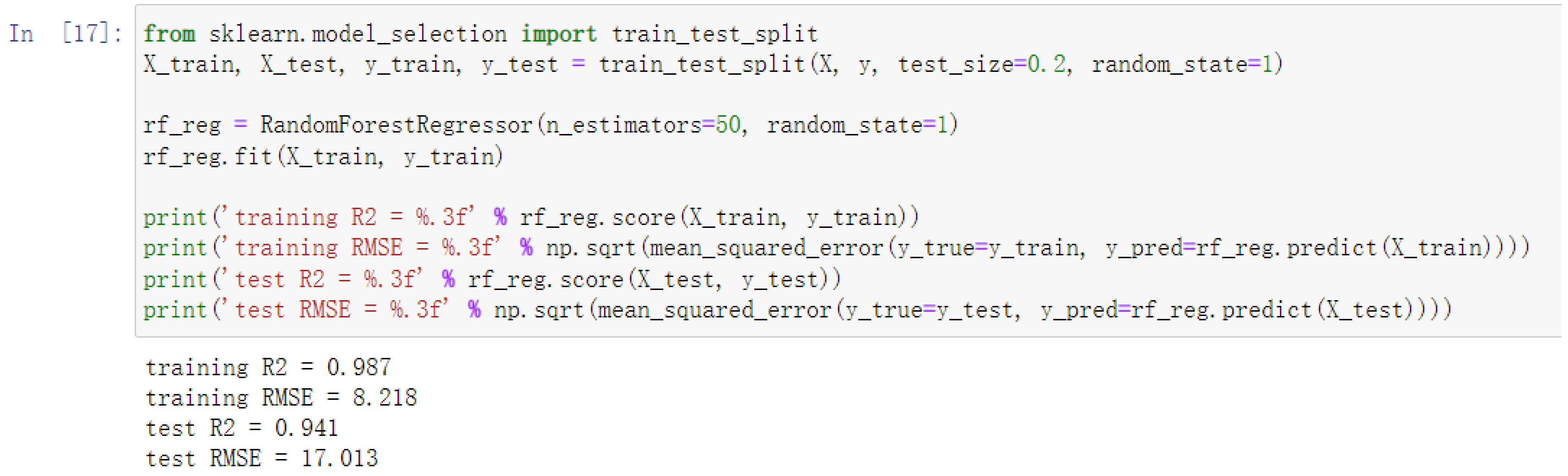
```
training R2 = 0.989
training RMSE = 7.669
```

随机森林模型

```
In  [16]: plt.figure(figsize=(5,5))
          plt.plot([0,400], [0, 400], 'r--')
          plt.scatter(y, rf.predict(X), s = 80, c = 'g', edgecolor = 'k', alpha = 0.7)

Out[16]: <matplotlib.collections.PathCollection at 0x267c649afa0>
```



留出法

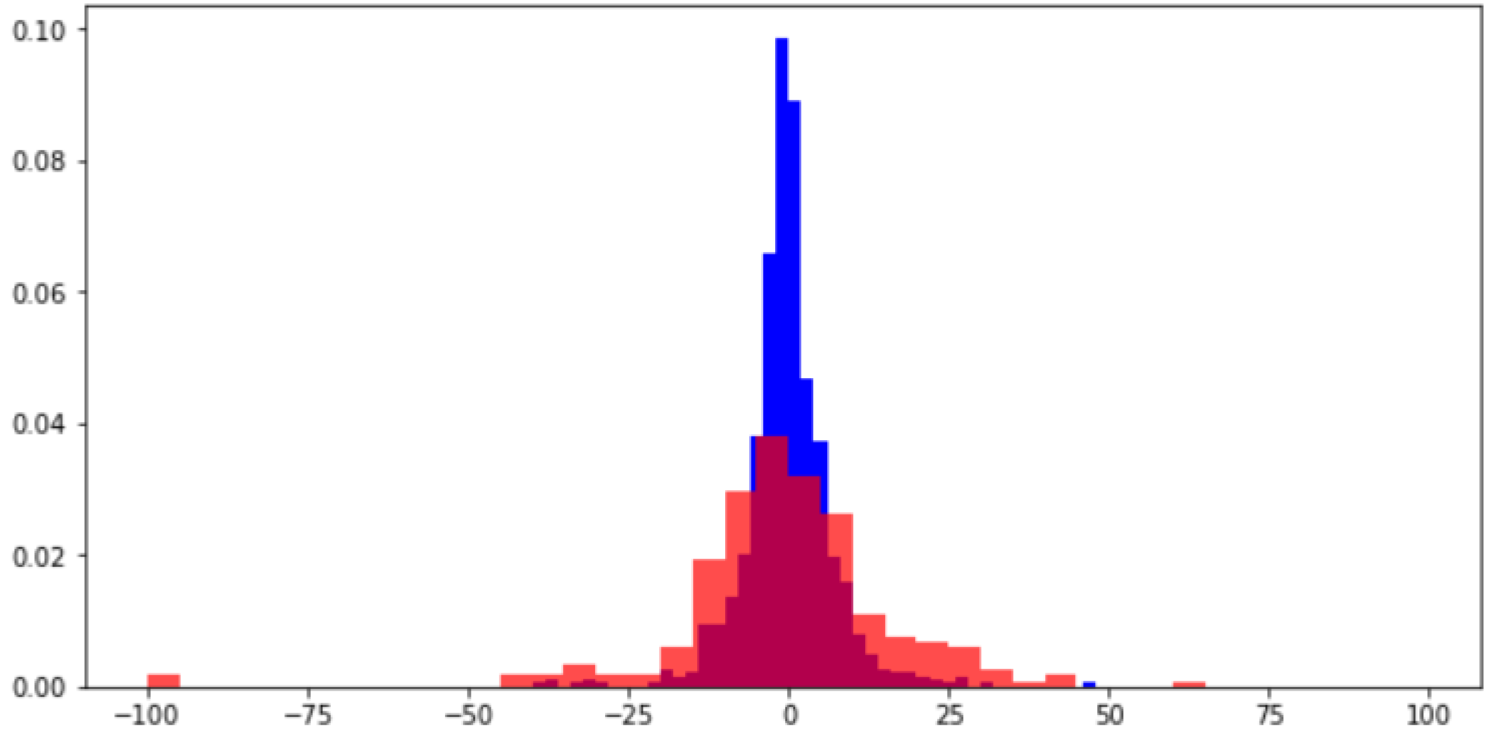


```
In [17]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

         rf_reg = RandomForestRegressor(n_estimators=50, random_state=1)
         rf_reg.fit(X_train, y_train)

         print('training R2 = %.3f' % rf_reg.score(X_train, y_train))
         print('training RMSE = %.3f' % np.sqrt(mean_squared_error(y_true=y_train, y_pred=rf_reg.predict(X_train))))
         print('test R2 = %.3f' % rf_reg.score(X_test, y_test))
         print('test RMSE = %.3f' % np.sqrt(mean_squared_error(y_true=y_test, y_pred=rf_reg.predict(X_test))))
```

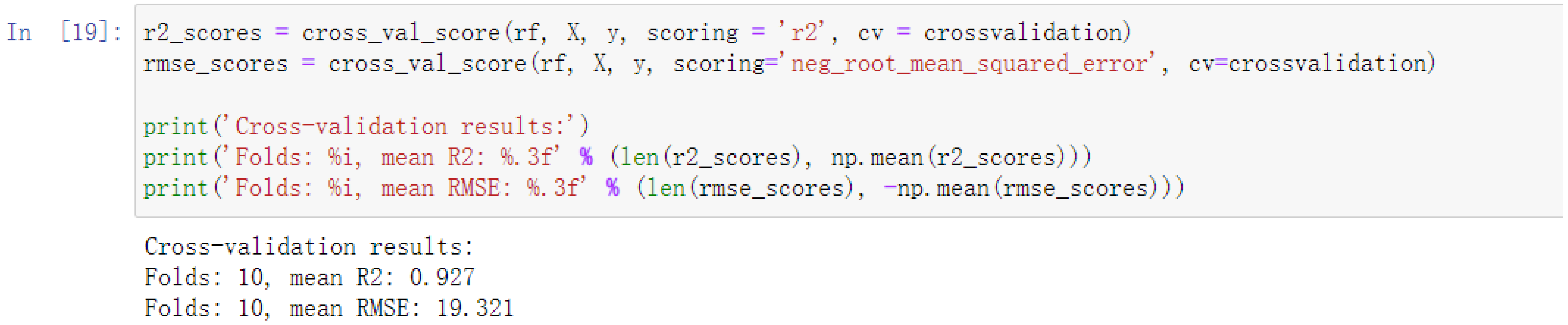
```
training R2 = 0.987
training RMSE = 8.218
test R2 = 0.941
test RMSE = 17.013
```

留出法

```
In  [18]: plt.figure(figsize=(10, 5))
          plt.hist(y_train-rf_reg.predict(X_train),color='blue',bins = np.arange(-100, 100, 2), density = True)
          plt.hist(y_test-rf_reg.predict(X_test),color='red', bins = np.arange(-100, 100, 5), density = True, alpha = 0.7)
          plt.show()
```



交叉验证

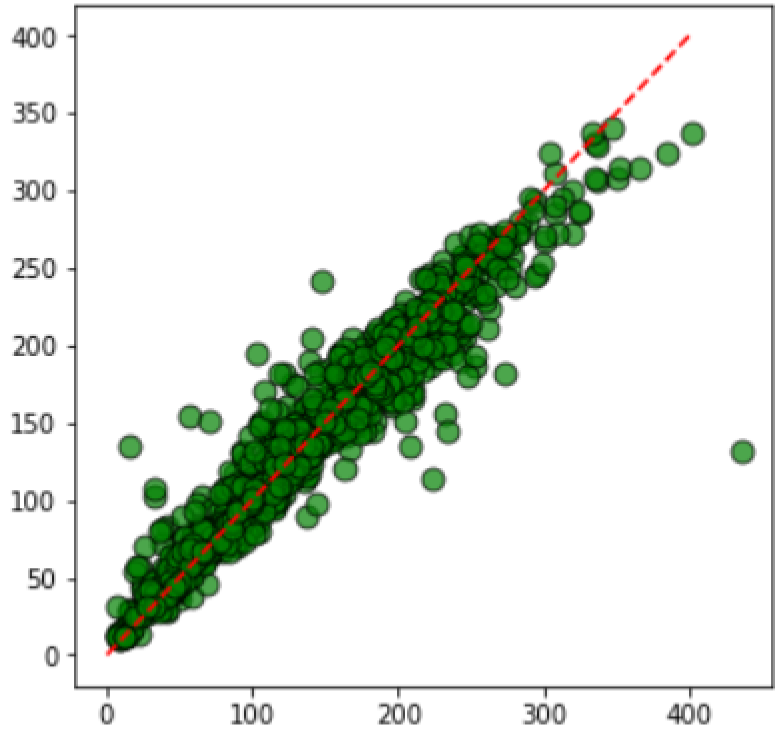


```
In  [19]: r2_scores = cross_val_score(rf, X, y, scoring = 'r2', cv = crossvalidation)
           rmse_scores = cross_val_score(rf, X, y, scoring='neg_root_mean_squared_error', cv=crossvalidation)

           print('Cross-validation results:')
           print('Folds: %i, mean R2: %.3f' % (len(r2_scores), np.mean(r2_scores)))
           print('Folds: %i, mean RMSE: %.3f' % (len(rmse_scores), -np.mean(rmse_scores)))
```

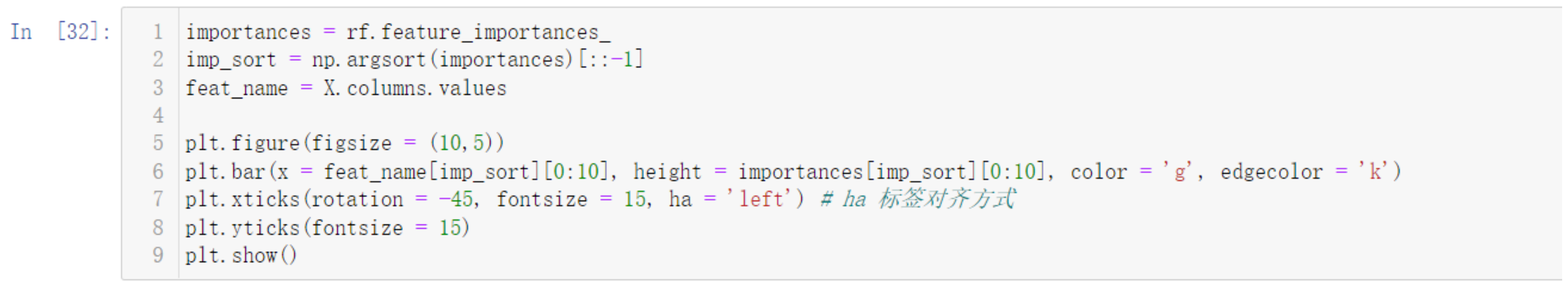
```
Cross-validation results:
Folds: 10, mean R2: 0.927
Folds: 10, mean RMSE: 19.321
```

交叉验证

```
In  [20]: plt.figure(figsize=(5,5))
          plt.plot([0,400], [0, 400], 'r--')
          plt.scatter(y, cross_val_predict(rf, X, y, cv=crossvalidation), s = 80, c = 'g', edgecolor = 'k', alpha = 0.7)

Out[20]: <matplotlib.collections.PathCollection at 0x267c76484f0>
```

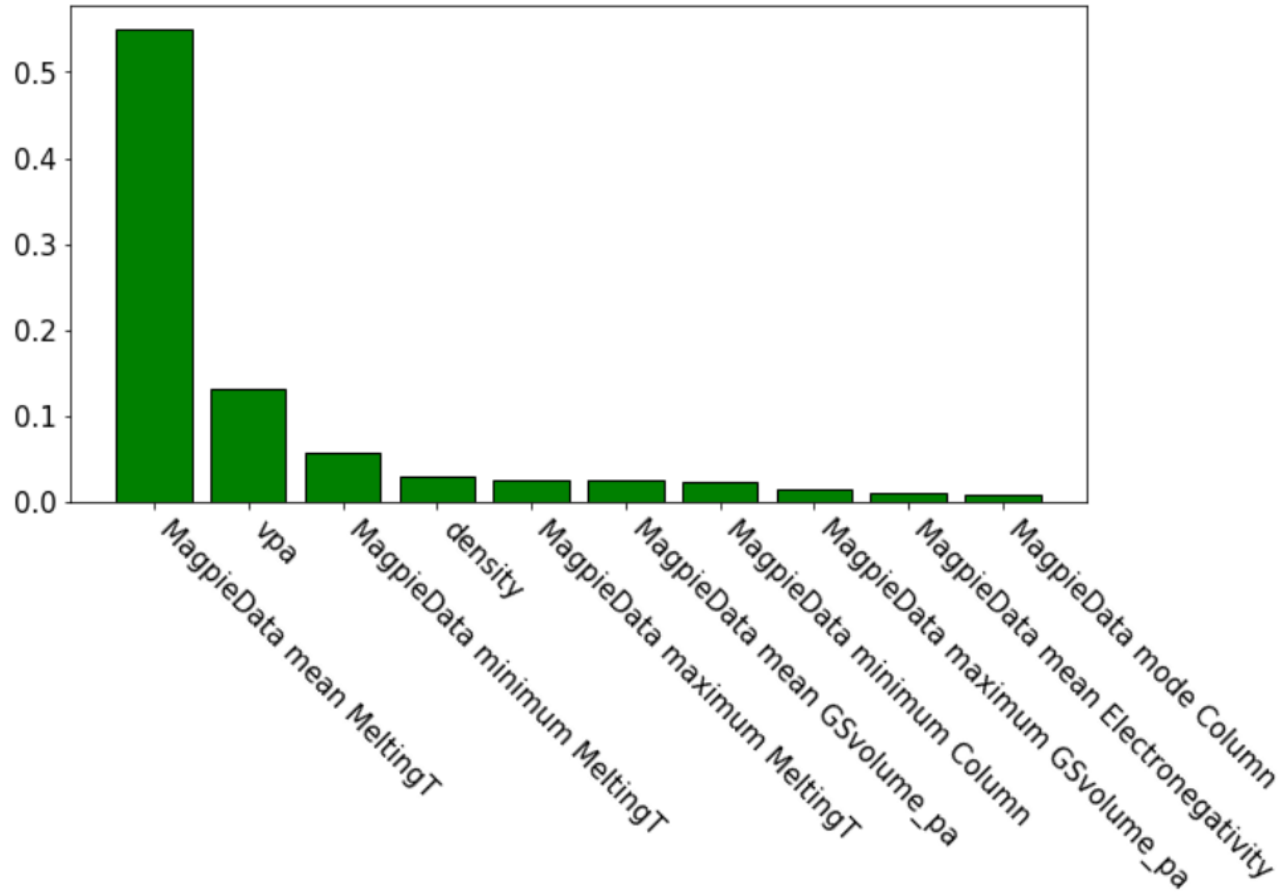


结果分析



In [32]:

```python
importances = rf.feature_importances_
imp_sort = np.argsort(importances)[::-1]
feat_name = X.columns.values

plt.figure(figsize = (10,5))
plt.bar(x = feat_name[imp_sort][0:10], height = importances[imp_sort][0:10], color = 'g', edgecolor = 'k')
plt.xticks(rotation = -45, fontsize = 15, ha = 'left') # ha 标签对齐方式
plt.yticks(fontsize = 15)
plt.show()
```


MagpieData mean MeltingT
vpa
MagpieData minimum MeltingT
density
MagpieData maximum MeltingT
MagpieData mean GSvolume_pa
MagpieData minimum Column
MagpieData maximum GSvolume_pa
MagpieData mean Electronegativity
MagpieData mode Column