

Regression-with-a-Mohs-Hardness-Dataset

～モース硬度データセットを使用した回帰～



EDAの結果のレポート



: 目次



概要

- 目的
- 評価指標



データの構成

- データの情報
- カラム名



EDA（探索的データ分析）

- データセットの要約統計量
- 欠損値
- データセットのプロット（分布）
- 目的変数



参考資料



概要

Let's start with the first set of slides

目標：

鉱物の性質からモース硬度を回帰
を用いて予測すること

“

<https://www.kaggle.com/competitions/playground-series-s3e25>



概要

1. アプローチ

- ◎ この研究は、著者が物理的および光学的特性から、天然素材を直接含んだ鉱物の組み合わせの特徴の性質を特徴づけるCRC Handbook45からデータベースを構築した。

2. データセット:

1. 実験的なモース硬度データと結晶クラス。
2. 物性と鉱物データベースから派生。
3. 369種類の鉱物からなるデータセット。



概要-2

1. 前処理:
 1. 同名鉱物の組成の組み合わせを調整。
 2. 一意の組成622鉱物のデータベース作成。
2. 検証:
 1. 独立した検証データセット。
 2. 51の合成単結晶データ。
 3. 結晶構造ごとに複数のサンプルを含む。
3. 特徴:
 1. 11の原子ディスクリプタで材料を表現。
 2. 元素特徴: 電子数、価電子数、原子番号、イオン化エネルギーなど。

評価指標：Median Absolute Error (MedAE)

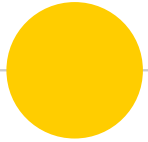
- ◎ 残差の絶対値の中央値
- ◎ MAEよりも更に外れ値に堅牢な評価指標

$$\begin{aligned}\text{MedAE} &= \text{median}(|\hat{y}_1 - y_1|, \dots, |\hat{y}_n - y_n|) \\ &= \text{中央値}(|\text{予測値}_{1\text{番目}} - \text{正解値}_{1\text{番目}}|, \dots, |\text{予測値}_{n\text{番目}} - \text{正解値}_{n\text{番目}}|)\end{aligned}$$

ここで、 \hat{y}^i は予測値であり、 y_i は各観測値 i の真の値

“

<https://atmarkit.itmedia.co.jp/ait/articles/2108/18/news023.html>



データの構成

Let's start with the first set of slides



データの情報

● データの形式

- データ数 : train(10407), test(6939)
- カラム数 : 11 + 目的変数 (Hardness)

● データの型

- 全てのカラムの値がfloat64 (浮動小数点) であった
 - カラム名 : object
- DataFrameの型: numpy.ndarray
- indexの型 : pandas.core.indexes.range.RangeIndex



カラム名

Hardness (目的変数)

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

allelectrons_Total

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

density_Total

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.

allelectrons_Average

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

val_e_Average

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

atomicweight_Average

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.



カラム名-2

ionenergy_Average

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

el_neg_chi_Average

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

R_vdw_element_Average

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.

R_cov_element_Average

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

zaratio_Average

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

density_Average

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.



Big concept

What's your EDA?

EDA（探索的データ分析）：

データセットに適宜**前処理**を施しつつ様々な統計量を抽出して可視化し、そこに内在する**特性・パターン・偏り**について探索的に仮説立案・検証を繰り返して分析すること



“



データセットの要約統計量

.describe()

Train

	count	mean	std	min	25%	50%	75%	max
allelectrons_Total	10407.0	128.053516	224.123776	0.0	68.000000	100.000000	131.000000	15300.000000
density_Total	10407.0	14.491342	15.972877	0.0	7.558488	10.650000	16.676996	643.093804
allelectrons_Average	10407.0	17.033222	10.468734	0.0	10.000000	12.600000	22.000000	67.000000
val_e_Average	10407.0	4.546789	0.690864	0.0	4.000000	4.714286	4.800000	6.000000
atomicweight_Average	10407.0	37.507703	26.012313	0.0	20.298893	26.203827	48.719500	167.400000
ionenergy_Average	10407.0	10.938308	1.408276	0.0	10.590660	11.202760	11.670725	15.245810
el_neg_chi_Average	10407.0	2.607662	0.334906	0.0	2.530000	2.706000	2.805000	3.443000
R_vdw_element_Average	10407.0	1.731330	0.192481	0.0	1.672500	1.732727	1.800000	2.250000
R_cov_element_Average	10407.0	0.944132	0.180017	0.0	0.864000	0.915556	0.981667	1.615840
zaratio_Average	10407.0	0.493349	0.063080	0.0	0.476196	0.488550	0.496070	0.825990
density_Average	10407.0	2.132984	1.936656	0.0	0.814800	1.351550	2.741550	10.970000
Hardness	10407.0	4.647126	1.680525	1.0	3.000000	5.500000	6.000000	10.000000

Test

	count	mean	std	min	25%	50%	75%	max
allelectrons_Total	6939.0	126.460128	207.564499	0.0	68.000000	100.000000	128.000000	10116.000000
density_Total	6939.0	14.794020	18.982447	0.0	7.558488	10.650000	16.601328	643.093804
allelectrons_Average	6939.0	17.406186	10.996089	0.0	10.000000	12.666667	22.000000	67.000000
val_e_Average	6939.0	4.546852	0.683158	0.0	4.000000	4.750000	4.800000	6.000000
atomicweight_Average	6939.0	38.422790	27.344351	0.0	20.298893	26.203827	48.719500	167.400000
ionenergy_Average	6939.0	10.921512	1.378980	0.0	10.584314	11.202760	11.645560	15.245810
el_neg_chi_Average	6939.0	2.608119	0.322873	0.0	2.527500	2.706000	2.806667	3.443000
R_vdw_element_Average	6939.0	1.737907	0.190584	0.0	1.678000	1.736000	1.820000	2.250000
R_cov_element_Average	6939.0	0.949638	0.179266	0.0	0.866667	0.920000	0.990000	1.615333
zaratio_Average	6939.0	0.491675	0.060829	0.0	0.476095	0.488550	0.496118	0.825990
density_Average	6939.0	2.152065	1.958213	0.0	0.812440	1.351550	2.780220	10.970000



データセットの要約統計量 カラムごとの分散

`np.var(, axis=0)`

Train

- allelectrons_Total 50226.640420
- density_Total 255.108273
- allelectrons_Average 109.583862
- val_e_Average 0.477247
- atomicweight_Average 676.575409
- ionenergy_Average 1.983050
- el_neg_chi_Average 0.112151
- R_vdw_element_Average 0.037045
- R_cov_element_Average 0.032403
- zaratio_Average 0.003979
- density_Average 3.750275
- Hardness 2.823894

Test

- allelectrons_Total 43076.812368
- density_Total 360.281362
- allelectrons_Average 120.896553
- val_e_Average 0.466638
- atomicweight_Average 747.605784
- ionenergy_Average 1.901313
- el_neg_chi_Average 0.104232
- R_vdw_element_Average 0.036317
- R_cov_element_Average 0.032132
- zaratio_Average 0.003700
- density_Average 3.834044



データセットの要約統計量 変動係数

変動係数：標準偏差を平均値で割った値

- スケールに依存せず，比較できるようになる

.std() / .mean()

Train

- allelectrons_Total 1.750235
- density_Total 1.102236
- allelectrons_Average 0.614607
- val_e_Average 0.151945
- atomicweight_Average 0.693519
- ionenergy_Average 0.128747
- el_neg_chi_Average 0.128431
- R_vdw_element_Average 0.111175
- R_cov_element_Average 0.190669
- zaratio_Average 0.127861
- density_Average 0.907956
- Hardness 0.361627



データセットの要約統計量 カラムごとの尖度・歪度

`.skew()`, `.kurt()`

◎ 尖度

- 分布が正規分布からどれだけ尖っているかを表す統計量で、山の尖り度と裾の広がり度

◎ 歪度

- 分布が正規分布からどれだけ歪んでいるかを表す統計量で、左右対称性を示す指標のこと

目的変数の尖度と歪度

尖度: -0.125749

歪度: -0.793775



データセットの要約統計量 カラムごとの歪度

.kurtosis()

Train

- allelectrons_Total 2383.425529
- density_Total 259.161522
- allelectrons_Average 2.740956
- val_e_Average 12.494568
- atomicweight_Average 3.332072
- ionenergy_Average 24.733284
- el_neg_chi_Average 22.175230
- R_vdw_element_Average 40.696122
- R_cov_element_Average 5.156704
- zaratio_Average 24.828071
- density_Average 2.422794
- Hardness -0.793775

Test

- allelectrons_Total 1556.276209
- density_Total 313.397231
- allelectrons_Average 2.668006
- val_e_Average 11.225967
- atomicweight_Average 3.227547
- ionenergy_Average 23.335279
- el_neg_chi_Average 20.556445
- R_vdw_element_Average 40.880686
- R_cov_element_Average 5.075099
- zaratio_Average 27.667758
- density_Average 2.115909



欠損値

すべてのカラムで欠損値が
確認されなかった

- 元々欠損値がないデータ

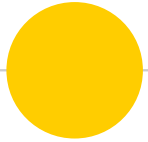
`.isnull().sum()`

Train

- allelectrons_Total 0
- density_Total 0
- allelectrons_Average 0
- val_e_Average 0
- atomicweight_Average 0
- ionenergy_Average 0
- el_neg_chi_Average 0
- R_vdw_element_Average 0
- R_cov_element_Average 0
- zaratio_Average 0
- density_Average 0
- Hardness 0

Train

- allelectrons_Total 0
- density_Total 0
- allelectrons_Average 0
- val_e_Average 0
- atomicweight_Average 0
- ionenergy_Average 0
- el_neg_chi_Average 0
- R_vdw_element_Average 0
- R_cov_element_Average 0
- zaratio_Average 0
- density_Average 0



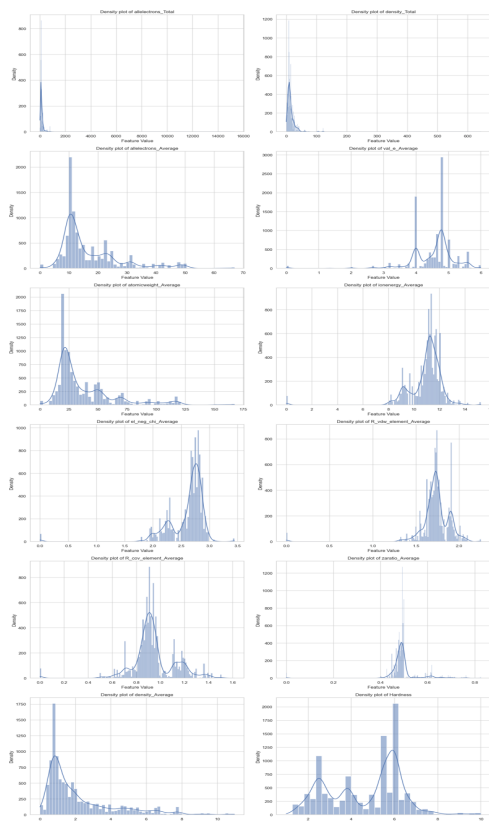
データセットのプロット (分布)

Let's start with the first set of slides

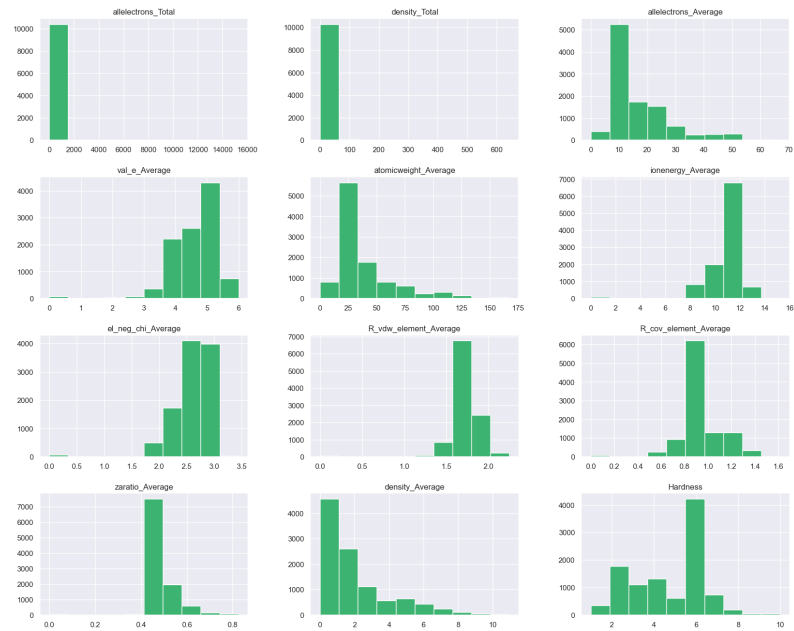


データセットのプロット (分布) .histplot()

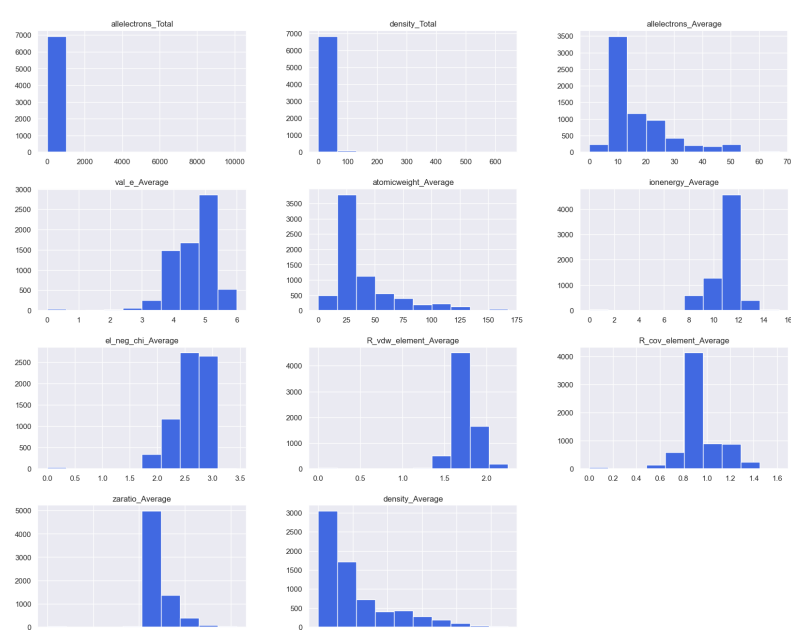
密度関数



Train



Test



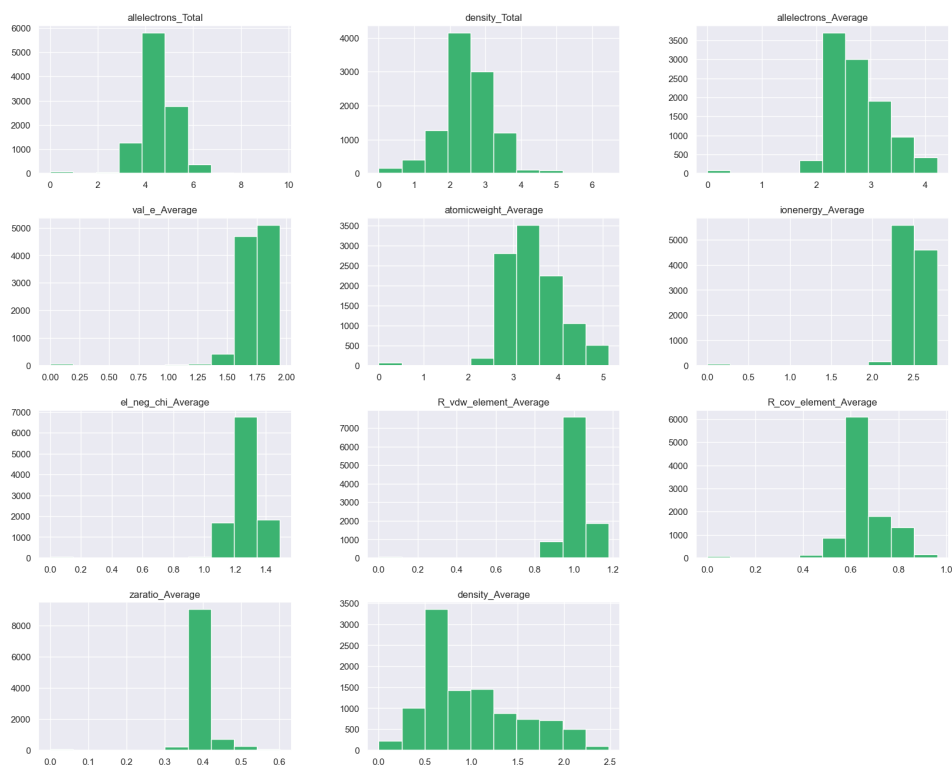


データセットのプロット (分布) 密度関数 (対数変換後)

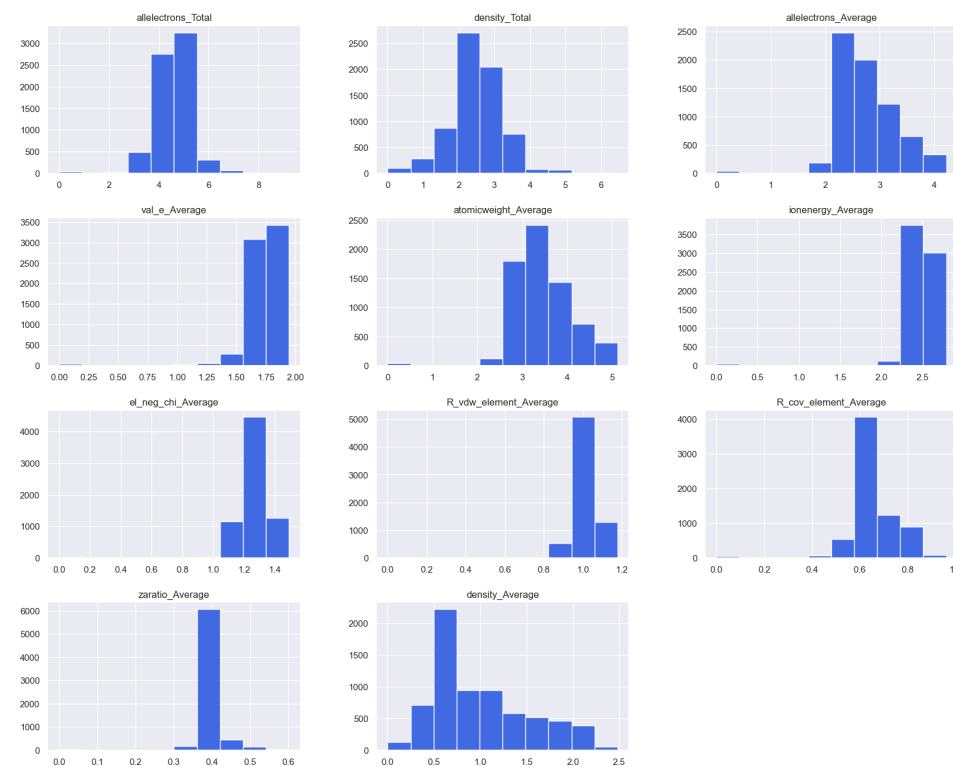
$\text{np.log}(x + 1)$, $x=\text{len}(\text{columns})$

分布が確認しやすくなった

Train



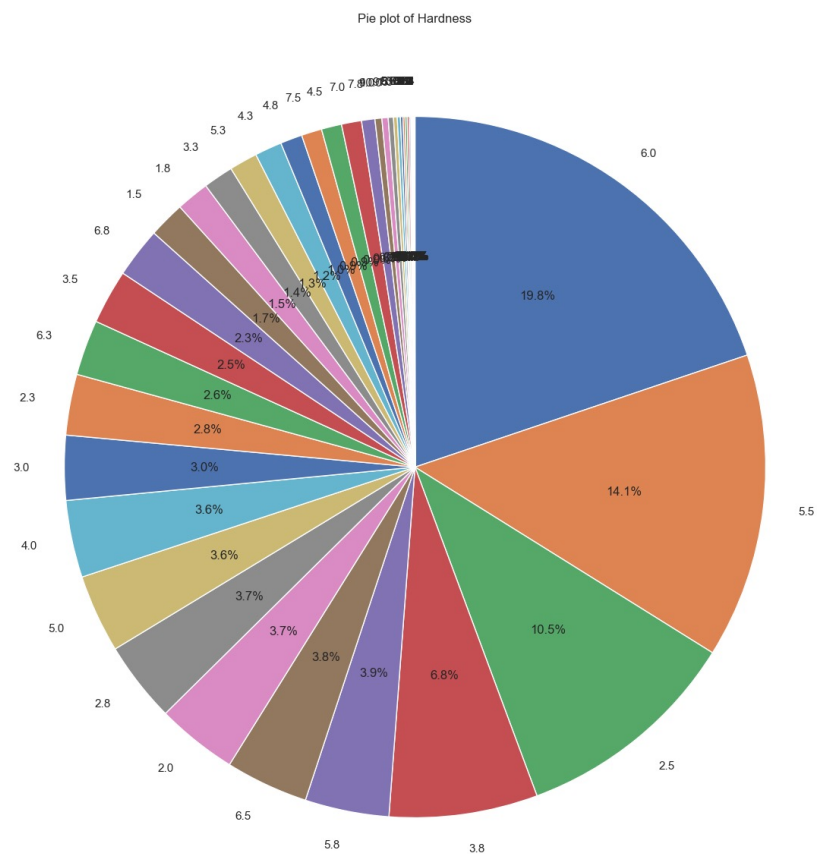
Test





円グラフ：目的変数 (Hardness)

plt.pie()



MOST FREQUENT VALUES

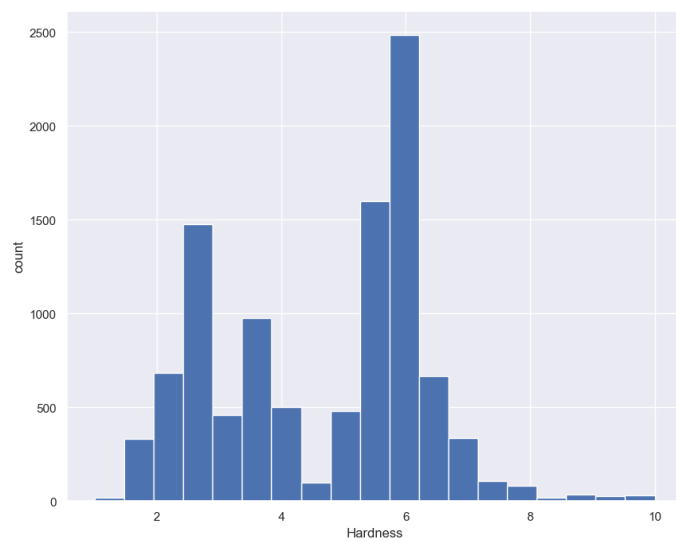
6.0	2,063	19.8%
5.5	1,463	14.1%
2.5	1,089	10.5%
3.8	712	6.8%
5.8	403	3.9%
6.5	397	3.8%
2.0	388	3.7%
2.8	387	3.7%
5.0	375	3.6%
4.0	370	3.6%
3.0	310	3.0%
2.3	292	2.8%
6.3	266	2.6%
3.5	261	2.5%
6.8	240	2.3%



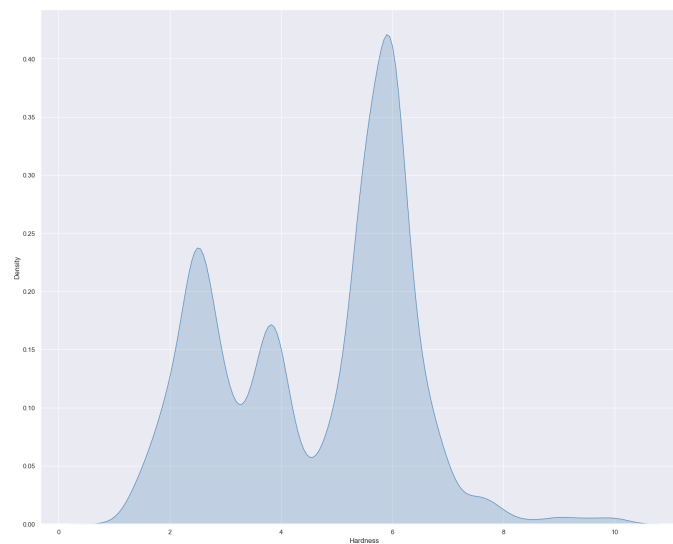
ヒストグラム：目的変数

`plt.hist()`

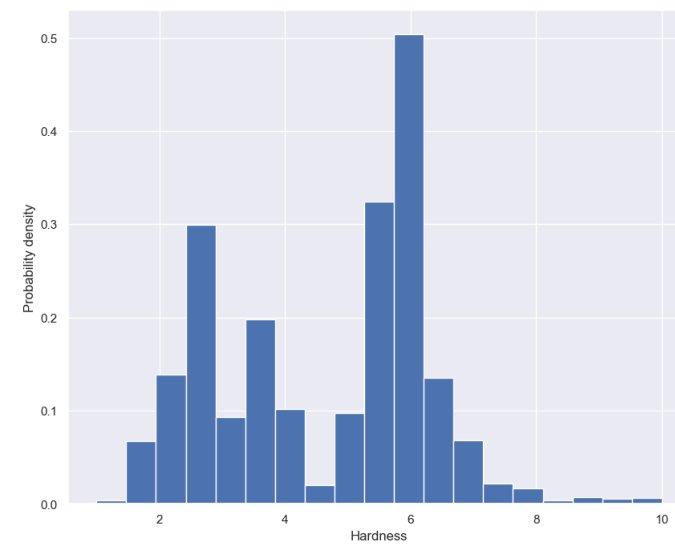
目的変数のどの数値が多いのか、少ないのか、偏りがあるのかを調べる



kdeplot

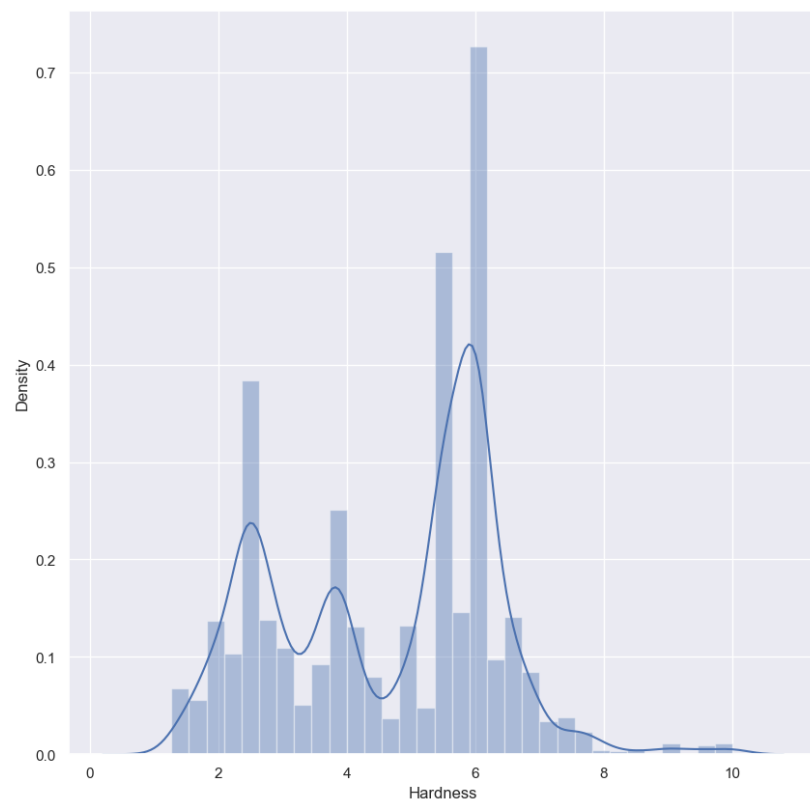


確率表現



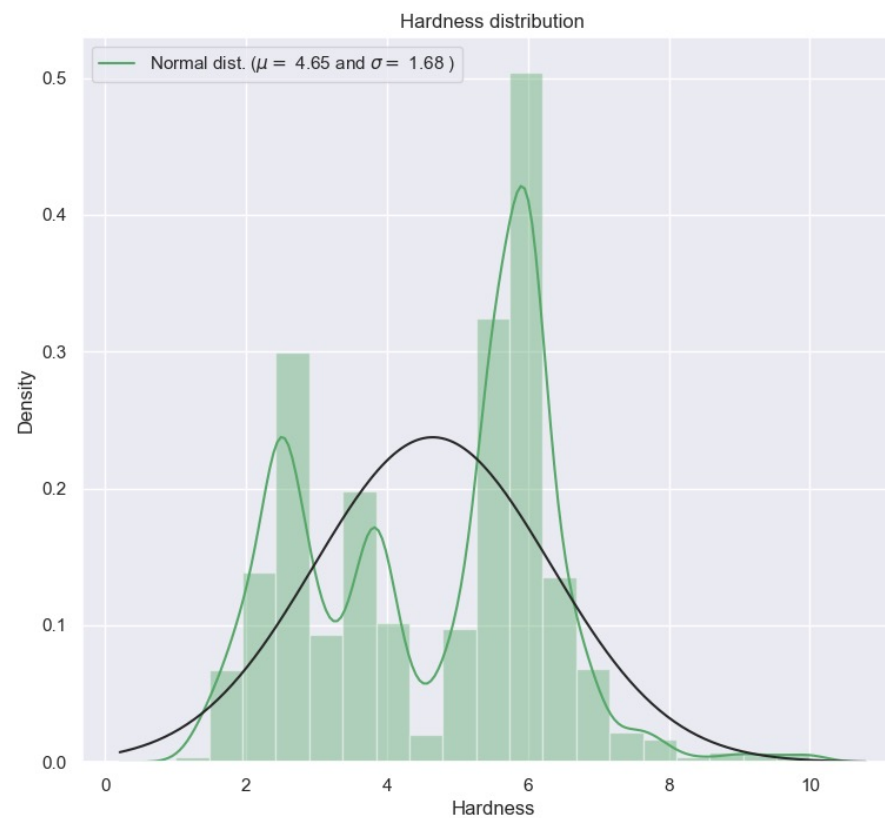


ヒストグラム：目的変数



`sns.distplot()`

正規分布付きのヒストグラム（MLE）



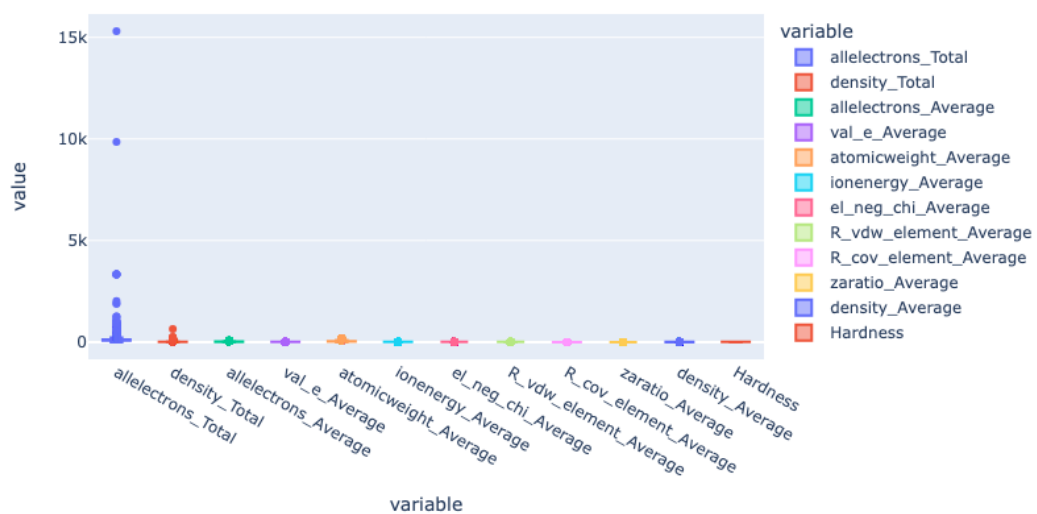


箱ひげ図

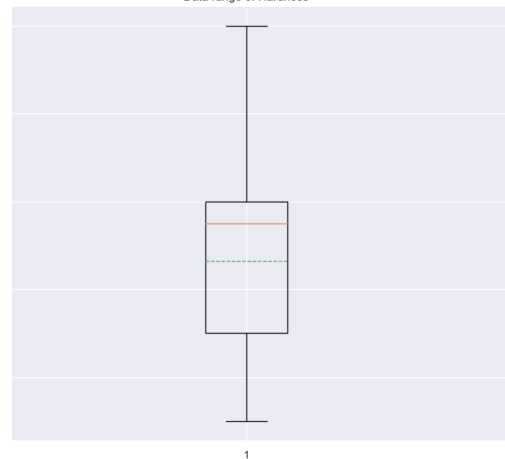
`px.box()`, `plt.boxplot()`, `plt.violinplot()`

データの範囲を調べる

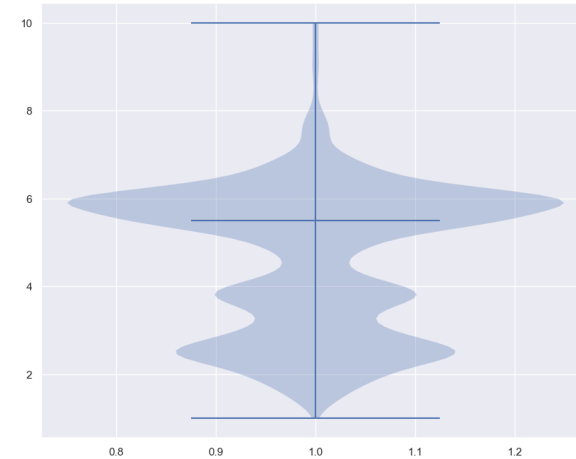
Box Plots



Data range of Hardness



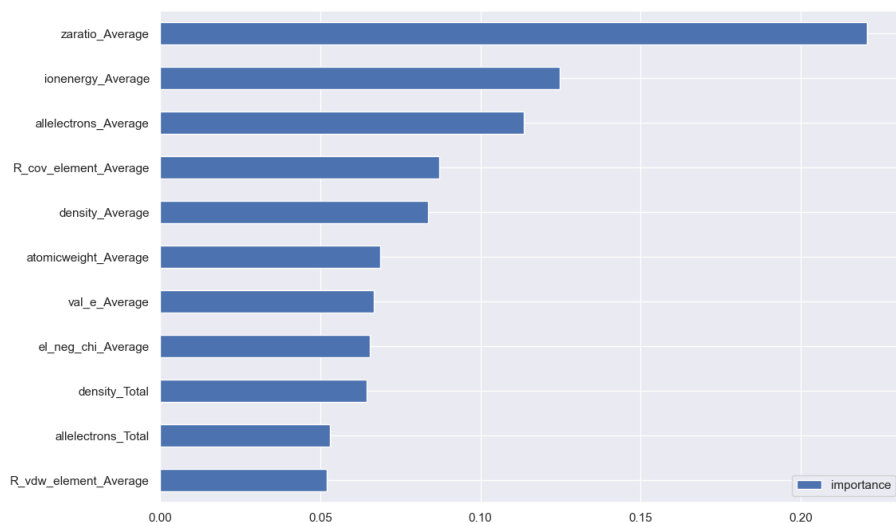
Violin plot of Hardness





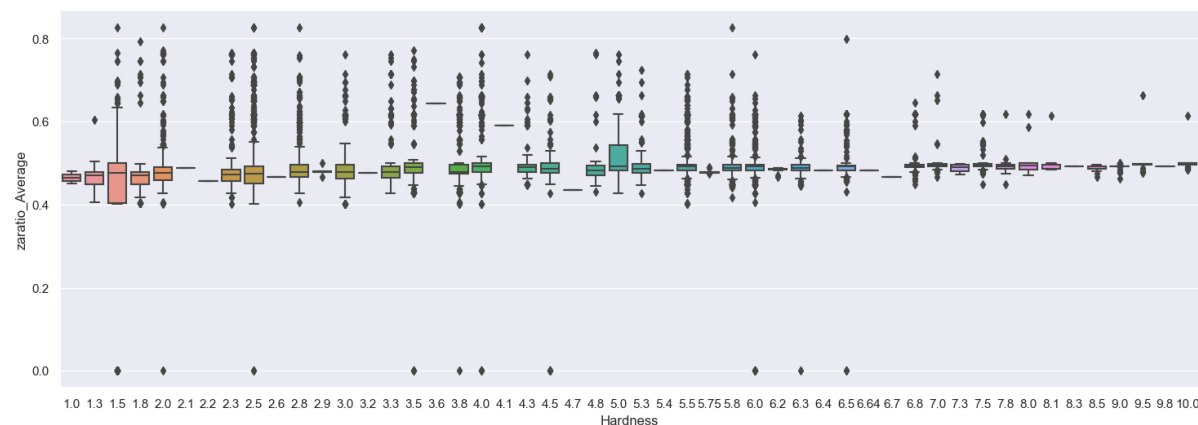
特徴量の重要度評価

'zaratio_Average'が重要度が特に高い



RandomForestRegressor(), sns.boxplot()

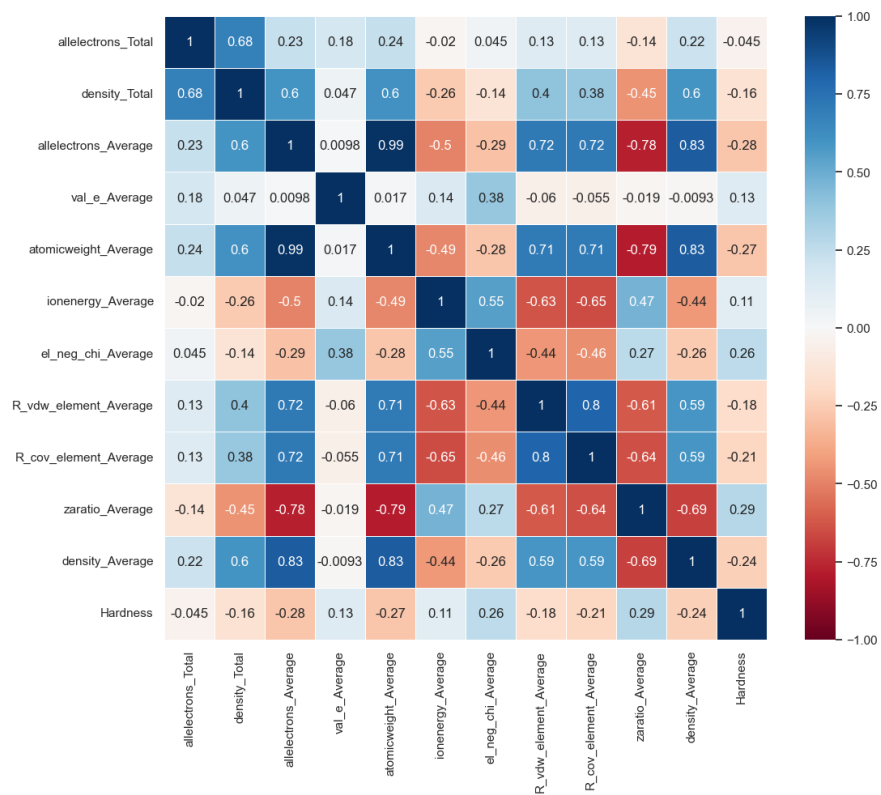
目的変数と'zaratio_Average'
(最も特徴重要度の高い説明変数) との関係





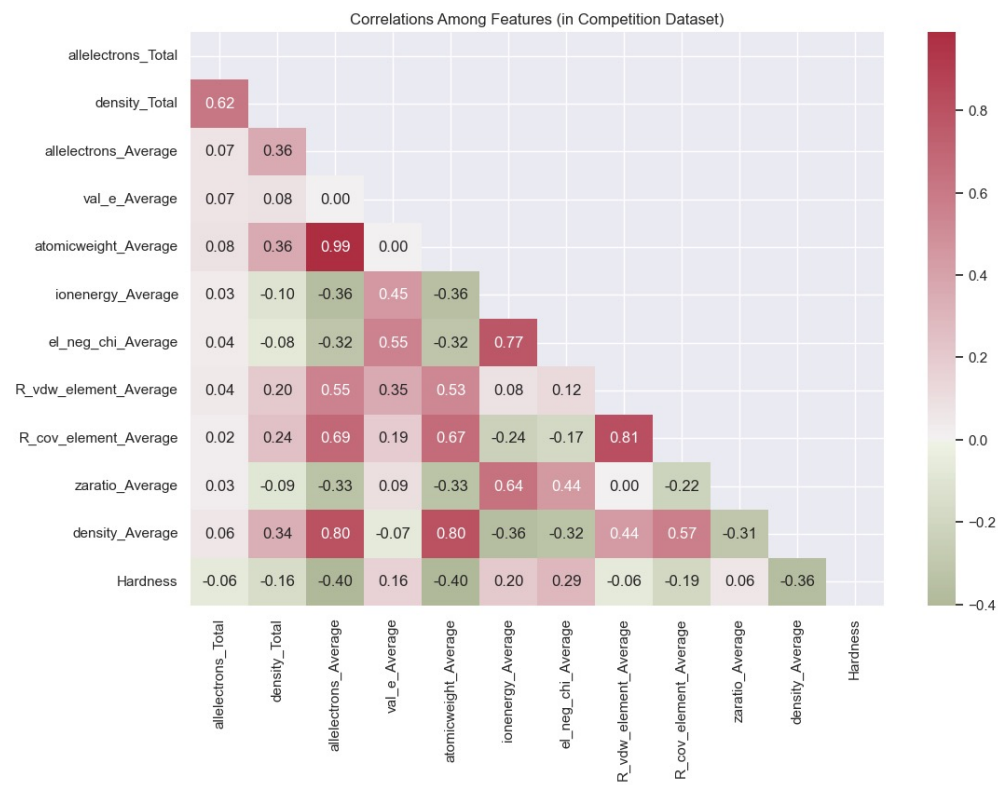
相関（ヒートマップ）

'allelectrons_Average'と'atomicweight_Average'の間には強い相関がある（99%の相関）



sns.heatmap()

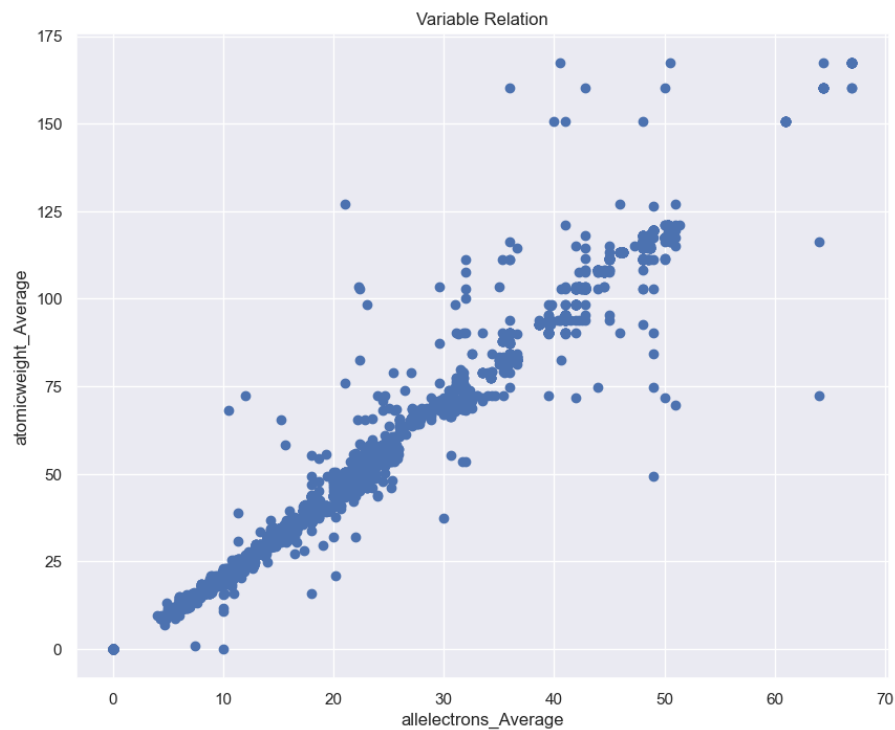
目的変数





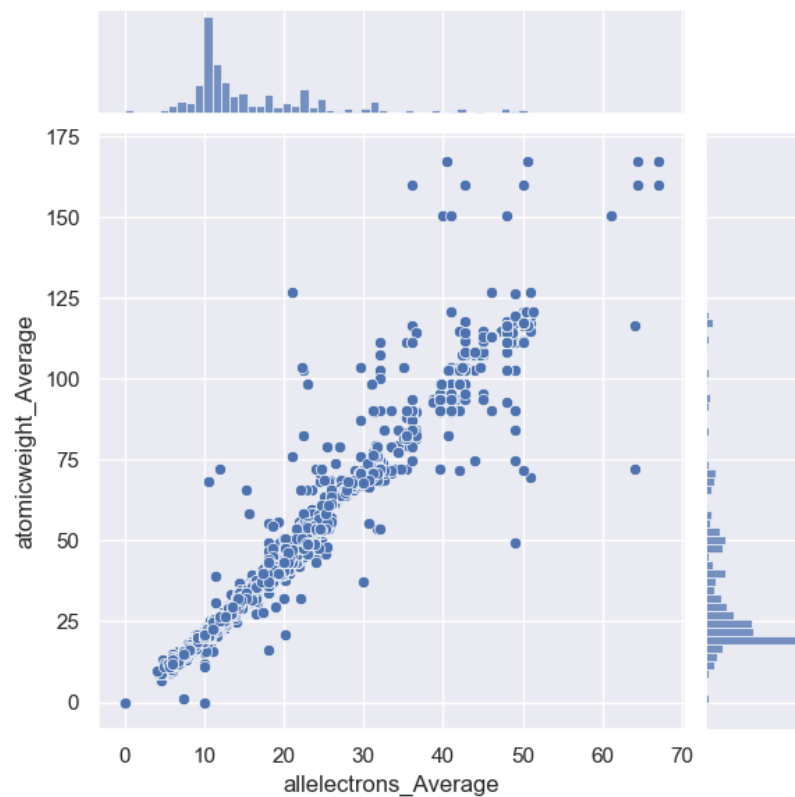
散布図

ヒートマップの結果から，散布図で変数の関係性で調べる



`plt.scatter()`

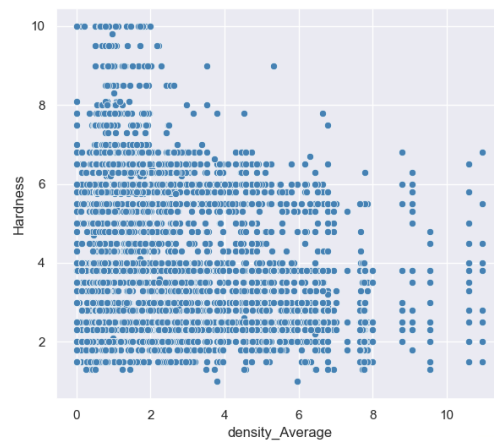
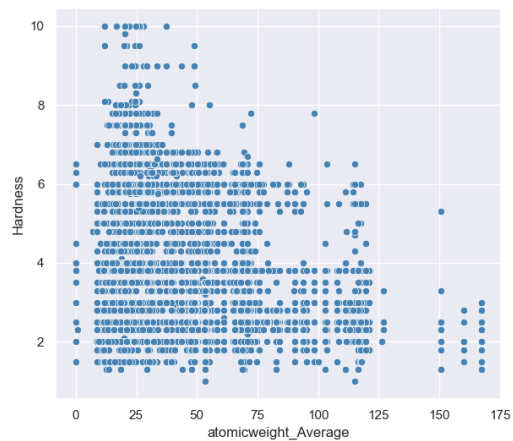
二変量グラフと単変量グラフを作成する



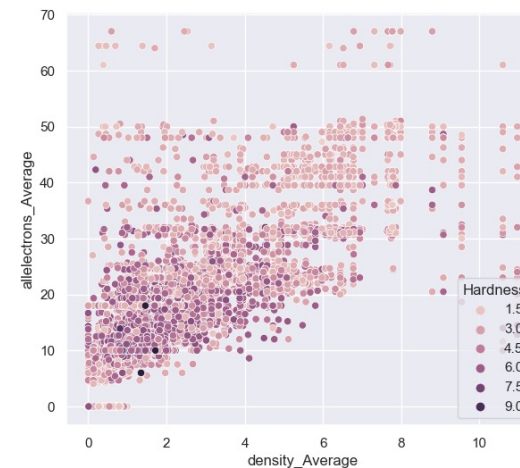
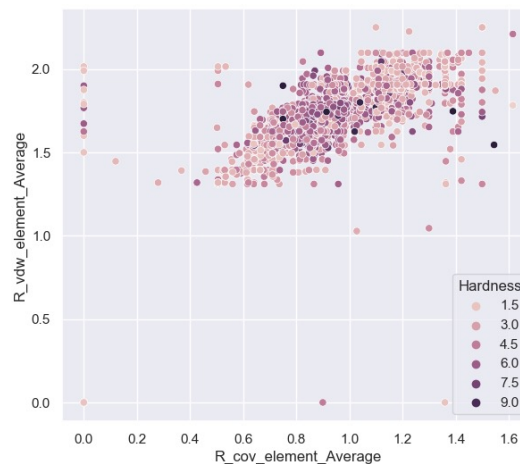


高い相関を持つ説明変数の散布図

plt.scatterplot()







右のグラフでは、プロットの上半分（allelectrosn_Average > 40）で、硬度（Hardness）が4.5以下



参考資料

- Pandasクックブック Pythonによるデータ処理のレシピ
- NumPy & SciPy数値計算実装ハンドブック数値シミュレーション入門者のための
- 東京大学のデータサイエンティスト育成講座 Pythonで手を動かして学ぶデータ分析
- Matplotlib & Seaborn実装ハンドブックより美しい統計データ可視化のための

Kaggle

-  Kapturov's solution of PS S3E25
- My Beauty Notebook 
- PS-S3-Ep25 | EDA  | Modeling + Submission 

記事

- sweetvizの記事