Binary Classification with a Software Defects Dataset

〜ソフトウェア欠陥データセットによるバイナリ分類〜

EDAの結果のレポート

TABLE OF CONTENTS（目次）

1. **概要** P.3
   目的，評価指標
2. **データの構成** P.7
   データの情報，カラム名
3. **EDA（探索的データ分析）** P.10
   要約統計量，欠損値，分布，目的変数
4. **参考資料** P.31

概要

Let’s start with the first set of slides

“

目的：

ソフトウェアの欠陥確率を予測する

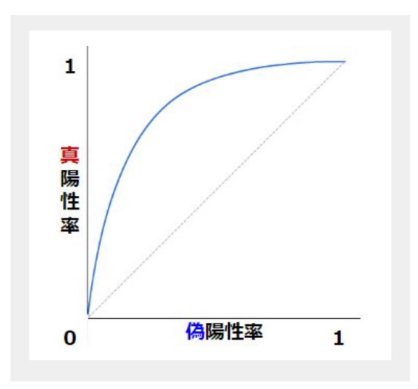
https://www.kaggle.com/competitions/playground-series-s3e23

概要

1. 目的：
   ソフトウェアの欠陥確率を予測する
2. データセット
   1. 全ての特徴量（説明変数）は数値型
   2. データセットのバランスがやや取れていない
      サンプルの23%のみが陽性に属している
3. 目的変数： defects(bool)

評価指標：　Area Under the ROC Curve(AUC)

- ROC曲線に基づくモデルの評価指標
- ROC曲線と横軸で囲まれる面積の値
- 大小比較を行い相対的な順番を与える
- 不均衡データ（imbalanced data）下におけるモデルの評価時にも耐えられる指標



データの構成

Let’s start with the first set of slides

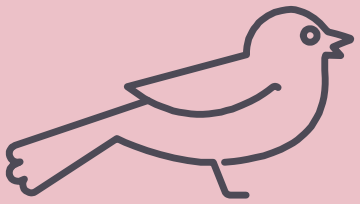
データの情報

- データの形式
  - データ数：train(101763), test(67842)
  - カラム数：21 + 目的変数（defects）
- データの型
  - int64: 17, float64: 4, bool: 1(目的変数)
    - 説明変数は数値（int64, float64）
  - DataFrameの型：numpy.ndarray
  - indexの型：pandas.core.indexes.range.RangeIndex

📖 各特徴量（カラム名）の日本語訳

データセットの9つのカラムは「算出された指標」

- P = volume = V = N * log2(mu) (長さNのプログラムを書くために必要な精神的比較の数)
- V* = minimal implementationでのvolume = (2 + mu2')*log2(2 + mu2’)
- L = プログラムの長さ = V*/N
- D = 難易度 = 1/L
- L' = 1/D
- I = 知能 = L'*V’
- E = プログラムを書くための労力 = V/L
- T = プログラムを書くのにかかる時間 = E/18 秒



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
loc	101763.000000	37.347160	54.600401	1.000000	13.000000	22.000000	42.000000	3442.000000
v(g)	101763.000000	5.492684	7.900855	1.000000	2.000000	3.000000	6.000000	404.000000
ev(g)	101763.000000	2.845022	4.631262	1.000000	1.000000	1.000000	3.000000	165.000000
iv(g)	101763.000000	3.498826	5.534541	1.000000	1.000000	2.000000	4.000000	402.000000
n	101763.000000	96.655995	171.147191	0.000000	25.000000	51.000000	111.000000	8441.000000
v	101763.000000	538.280956	1270.791601	0.000000	97.670000	232.790000	560.250000	80843.080000
l	101763.000000	0.111634	0.100096	0.000000	0.050000	0.090000	0.150000	1.000000
d	101763.000000	13.681881	14.121306	0.000000	5.600000	9.820000	18.000000	418.200000
i	101763.000000	27.573007	22.856742	0.000000	15.560000	23.360000	34.340000	569.780000
e	101763.000000	20853.589876	190571.405427	0.000000	564.730000	2256.230000	10193.240000	16846621.120000
b	101763.000000	0.179164	0.421844	0.000000	0.030000	0.080000	0.190000	26.950000
t	101763.000000	1141.357982	9862.795472	0.000000	31.380000	125.400000	565.920000	935923.390000
lOCode	101763.000000	22.802453	38.541010	0.000000	7.000000	14.000000	26.000000	2824.000000
lOComment	101763.000000	1.773945	5.902412	0.000000	0.000000	0.000000	1.000000	344.000000
lOBlank	101763.000000	3.979865	6.382358	0.000000	1.000000	2.000000	5.000000	219.000000
locCodeAndComment	101763.000000	0.196604	0.998906	0.000000	0.000000	0.000000	0.000000	43.000000
uniq_Op	101763.000000	11.896131	6.749549	0.000000	8.000000	11.000000	16.000000	410.000000
uniq_Opnd	101763.000000	15.596671	18.064261	0.000000	7.000000	12.000000	20.000000	1026.000000
total_Op	101763.000000	57.628116	104.537660	0.000000	15.000000	30.000000	66.000000	5420.000000
total_Opnd	101763.000000	39.249698	71.692309	0.000000	10.000000	20.000000	45.000000	3021.000000
branchCount	101763.000000	9.839549	14.412769	1.000000	3.000000	5.000000	11.000000	503.000000

	count	nunique	%unique	null	type
loc	101763	378	0.371451	0	float64
v(g)	101763	106	0.104164	0	float64
ev(g)	101763	71	0.069770	0	float64
iv(g)	101763	84	0.082545	0	float64
n	101763	836	0.821517	0	float64
v	101763	4515	4.436780	0	float64
l	101763	55	0.054047	0	float64
d	101763	3360	3.301789	0	float64
i	101763	5171	5.081415	0	float64
e	101763	8729	8.577774	0	float64
b	101763	315	0.309543	0	float64
t	101763	8608	8.458870	0	float64
lOCode	101763	298	0.292837	0	int64
lOComment	101763	91	0.089423	0	int64
lOBlank	101763	94	0.092371	0	int64
locCodeAndComment	101763	29	0.028498	0	int64
uniq_Op	101763	70	0.068787	0	float64
uniq_Opnd	101763	176	0.172951	0	float64
total_Op	101763	623	0.612207	0	float64
total_Opnd	101763	485	0.476598	0	float64
branchCount	101763	144	0.141505	0	float64
defects	101763	2	0.001965	0	bool

データセットの要約統計量

.describe()

Test

	count	mean	std	min	25%	50%	75%	max
loc	67842.000000	37.855945	55.793918	1.000000	13.000000	22.000000	42.000000	1891.000000
v(g)	67842.000000	5.559462	8.237389	1.000000	2.000000	3.000000	6.000000	470.000000
ev(g)	67842.000000	2.857699	4.666370	1.000000	1.000000	1.000000	3.000000	165.000000
iv(g)	67842.000000	3.535715	5.339085	1.000000	1.000000	2.000000	4.000000	402.000000
n	67842.000000	98.222178	179.123492	0.000000	25.000000	51.000000	111.000000	8441.000000
v	67842.000000	548.422497	1318.416828	0.000000	96.210000	232.790000	560.250000	80843.080000
l	67842.000000	0.111521	0.100840	0.000000	0.040000	0.090000	0.140000	1.000000
d	67842.000000	13.762472	14.487425	0.000000	5.500000	9.820000	18.000000	418.200000
i	67842.000000	27.635377	23.420521	0.000000	15.540000	23.230000	34.270000	569.780000
e	67842.000000	22164.320048	207276.970847	0.000000	562.940000	2276.780000	10112.427500	16846621.120000
b	67842.000000	0.182075	0.418493	0.000000	0.030000	0.080000	0.190000	18.380000
t	67842.000000	1218.682009	10739.081373	0.000000	31.270000	126.490000	561.802500	935923.390000
lOCode	67842.000000	23.131924	39.606659	0.000000	7.000000	14.000000	26.000000	1599.000000
lOComment	67842.000000	1.832493	6.229052	0.000000	0.000000	0.000000	1.000000	344.000000
lOBlank	67842.000000	4.020739	7.136069	0.000000	1.000000	2.000000	5.000000	447.000000
locCodeAndComment	67842.000000	0.200834	1.044581	0.000000	0.000000	0.000000	0.000000	42.000000
uniq_Op	67842.000000	11.909776	6.974254	0.000000	8.000000	12.000000	16.000000	411.000000
uniq_Opnd	67842.000000	15.678370	17.979368	0.000000	7.000000	12.000000	20.000000	1026.000000
total_Op	67842.000000	58.325212	105.491447	0.000000	15.000000	30.000000	66.000000	3368.000000
total_Opnd	67842.000000	39.845258	72.756320	0.000000	10.000000	20.000000	44.000000	2301.000000
branchCount	67842.000000	9.985923	15.342975	1.000000	3.000000	5.000000	11.000000	826.000000

	count	nunique	%unique	null	type
loc	67842	371	0.546859	0	float64
v(g)	67842	96	0.141505	0	float64
ev(g)	67842	67	0.098759	0	float64
iv(g)	67842	73	0.107603	0	float64
n	67842	828	1.220483	0	float64
v	67842	4269	6.292562	0	float64
l	67842	54	0.079597	0	float64
d	67842	3106	4.578285	0	float64
i	67842	4855	7.156334	0	float64
e	67842	8160	12.027947	0	float64
b	67842	310	0.456944	0	float64
t	67842	7969	11.746411	0	float64
lOCode	67842	296	0.436308	0	float64
lOComment	67842	83	0.122343	0	float64
lOBlank	67842	98	0.144453	0	float64
locCodeAndComment	67842	26	0.038324	0	float64
uniq_Op	67842	68	0.100233	0	float64
uniq_Opnd	67842	174	0.256478	0	float64
total_Op	67842	598	0.881460	0	float64
total_Opnd	67842	468	0.689838	0	float64
branchCount	67842	138	0.203414	0	float64

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

loc	2.981175e+03
v(g)	6.242290e+01
ev(g)	2.144837e+01
iv(g)	3.063084e+01
n	2.929107e+04
v	1.614895e+06
l	1.001909e-02
d	1.994093e+02
i	5.224255e+02
e	3.631710e+10
b	1.779509e-01
t	9.727378e+07
lOCode	1.485395e+03
lOComment	3.483812e+01
lOBlank	4.073410e+01
locCodeAndComment	9.978042e-01
uniq_Op	4.555597e+01
uniq_Opnd	3.263143e+02
total_Op	1.092802e+04
total_Opnd	5.139737e+03
branchCount	2.077259e+02
defects	1.752766e-01
dtype: float64	

Test

loc	3.112915e+03
v(g)	6.785358e+01
ev(g)	2.177469e+01
iv(g)	2.850541e+01
n	3.208475e+04
v	1.738197e+06
l	1.016855e-02
d	2.098824e+02
i	5.485127e+02
e	4.296311e+10
b	1.751340e-01
t	1.153262e+08
lOCode	1.568664e+03
lOComment	3.880051e+01
lOBlank	5.092273e+01
locCodeAndComment	1.091133e+00
uniq_Op	4.863950e+01
uniq_Opnd	3.232529e+02
total_Op	1.112828e+04
total_Opnd	5.293404e+03
branchCount	2.354034e+02
dtype: float64	

データセットの要約統計量
変動係数

.std() / .mean()

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）

Train

```
loc 1.461969
v(g) 1.438432
ev(g) 1.627847
iv(g) 1.581828
n 1.770684
v 2.360833
l 0.896647
d 1.032117
i 0.828954
e 9.138542
b 2.354515
t 8.641281
lOCode 1.690213
lOComment 3.327279
lOBlank 1.603662
locCodeAndComment 5.080807
uniq_Op 0.567373
uniq_Opnd 1.158213
total_Op 1.814004
total_Opnd 1.826570
branchCount 1.464779
defects 1.847223
dtype: float64
```

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: 1.305878

歪度: -0.294689

データセットの要約統計量
カラムごとの歪度

.kurtosis()

Train

loc	352.165118
v(g)	501.426262
ev(g)	96.037809
iv(g)	1564.108781
n	209.573661
v	518.212614
l	5.031947
d	92.422268
i	57.408889
e	2887.029117
b	533.051549
t	2780.016981
lOCode	633.778523
lOComment	353.062903
lOBlank	166.023713
locCodeAndComment	220.287364
uniq_Op	274.160956
uniq_Opnd	502.419354
total_Op	264.104622
total_Opnd	225.724304
branchCount	179.053243
defects	-0.294689

dtype: float64

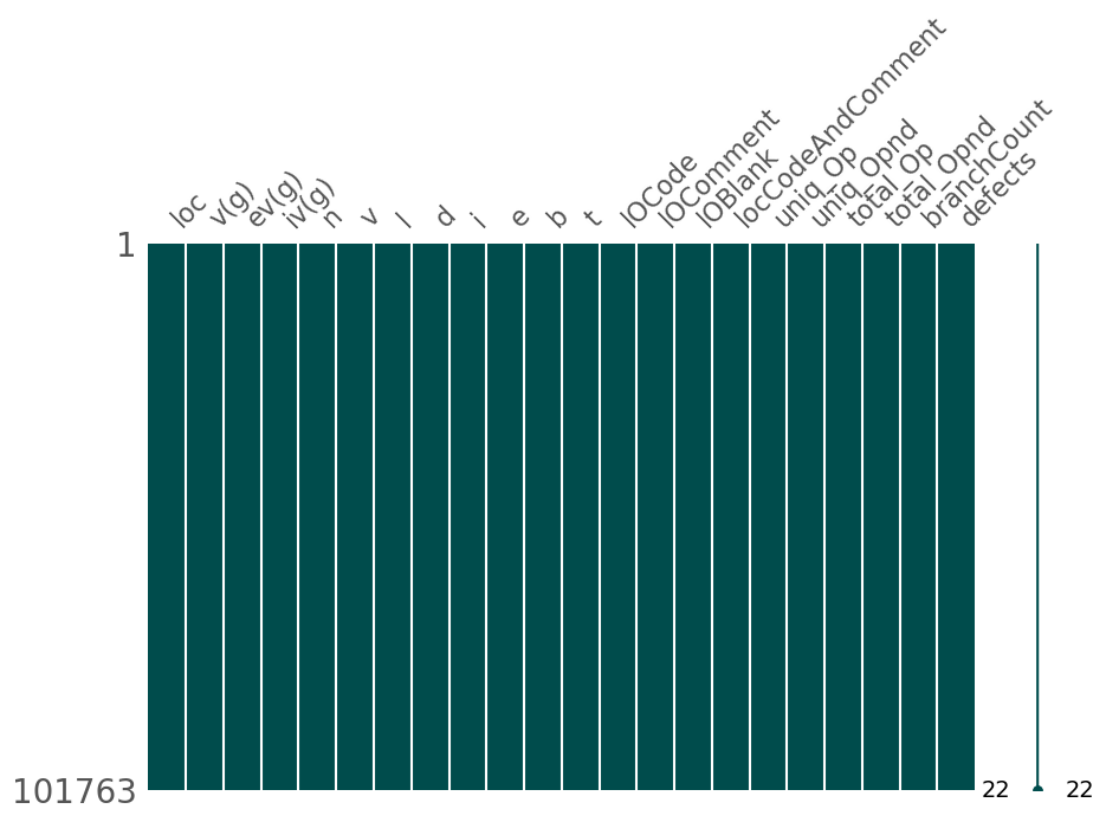
Test

loc	189.518138
v(g)	506.805542
ev(g)	93.199325
iv(g)	1199.446578
n	214.368472
v	521.219701
l	5.333381
d	85.792391
i	66.808480
e	2743.447235
b	325.059833
t	2560.566042
lOCode	321.081713
lOComment	386.809335
lOBlank	613.479470
locCodeAndComment	294.906326
uniq_Op	345.195385
uniq_Opnd	390.672954
total_Op	172.392128
total_Opnd	143.443636
branchCount	289.184268

dtype: float64

欠損値

.isnull().sum(), msno.matrix()



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

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） .hist()
密度関数

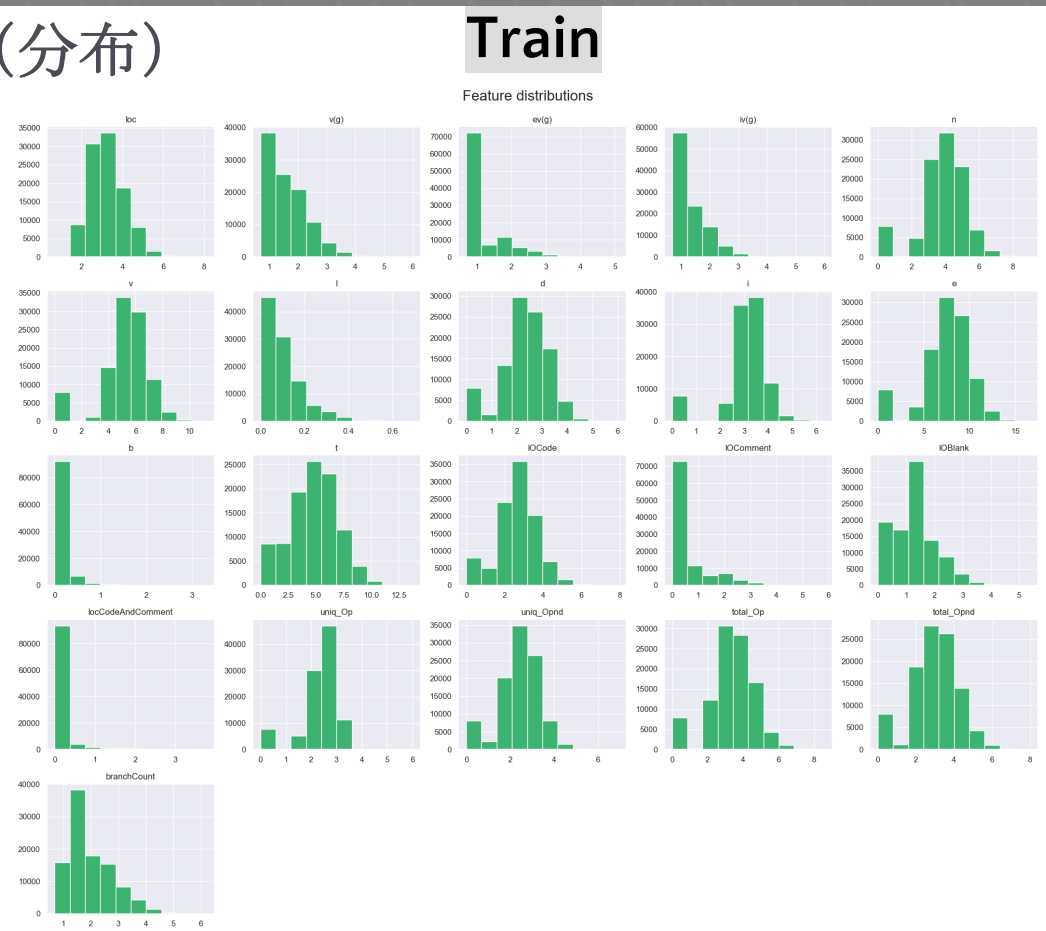
Train　分布がわかりにくい！！！

Test

データセットのプロット（分布）密度関数（対数変換後）

np.log(x + 1), x=len(columns)

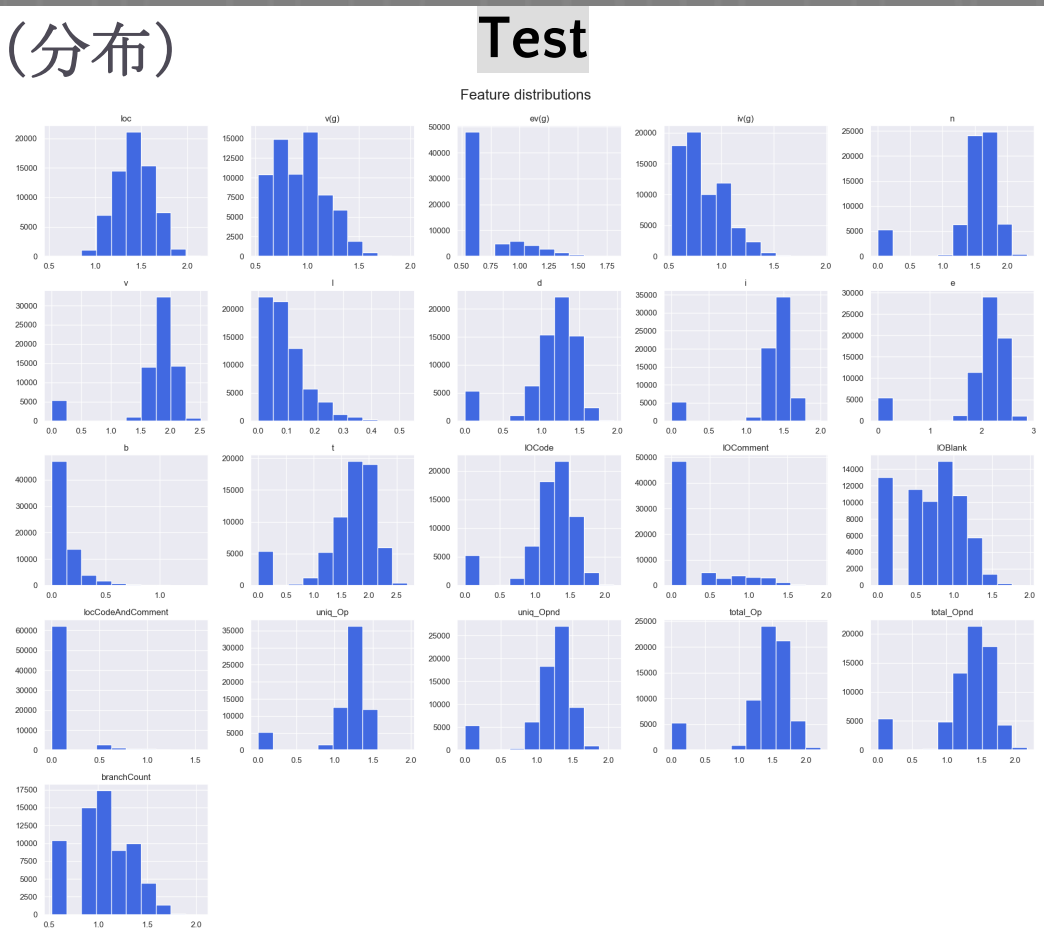
- 全ての説明変数は正の値
- ヒストグラムは左に偏っている
  - 指数分布に類似している



データセットのプロット（分布）密度関数（対数変換後）

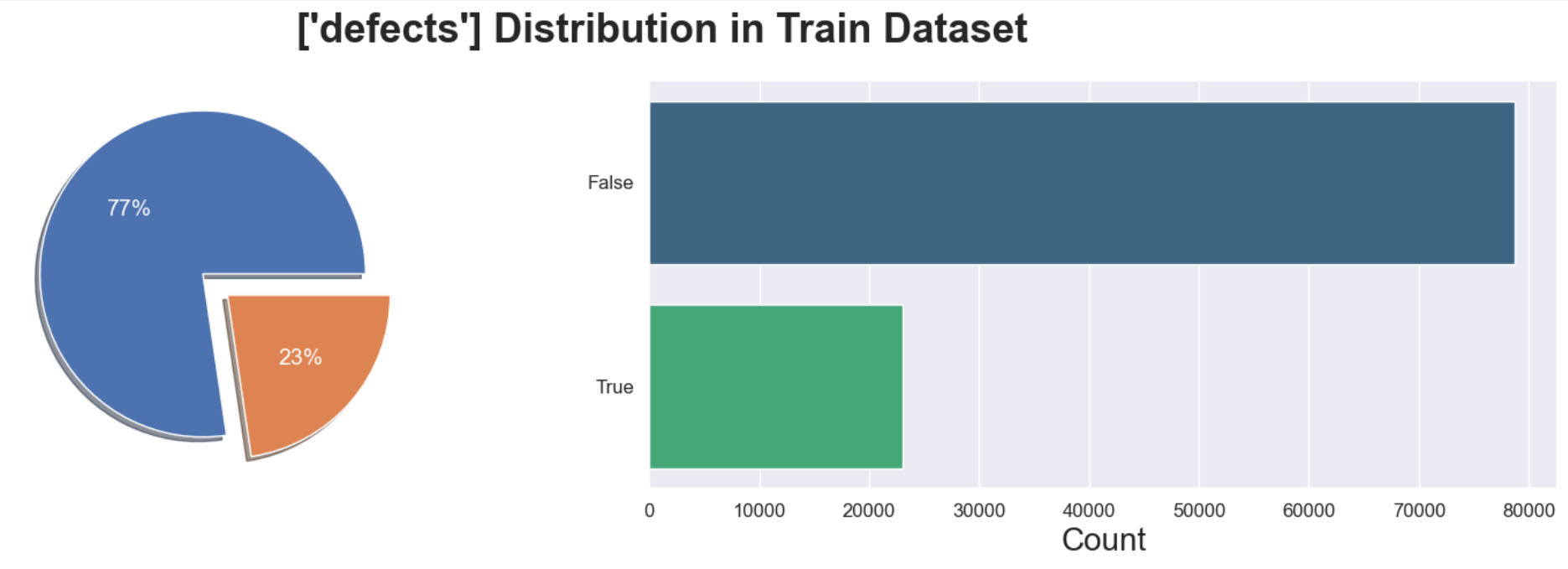
np.log(x + 1), x=len(columns)

- 一部の説明変数において分布が確認しやすくなった
- このような分布の場合，FunctionTransformer(np.log1p)でデータを対数変換すると，多くのモデルが恩恵を受ける可能性がある



円グラフ：目的変数

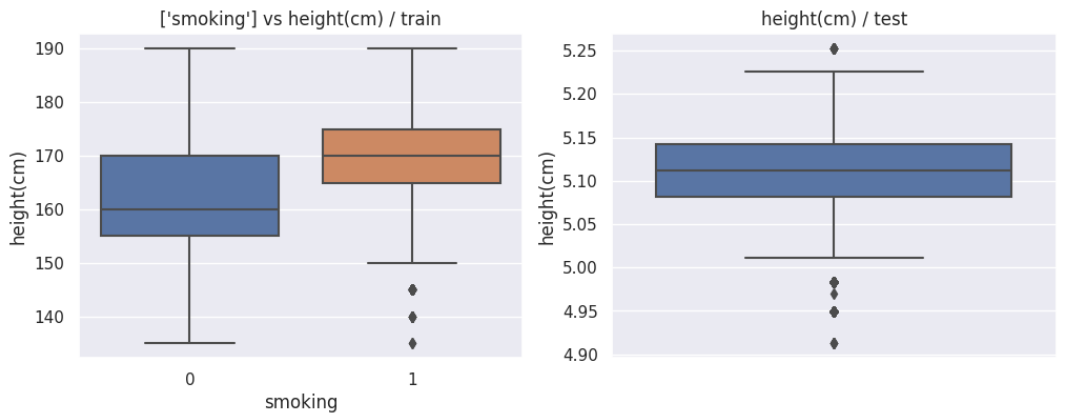
plt.pie()

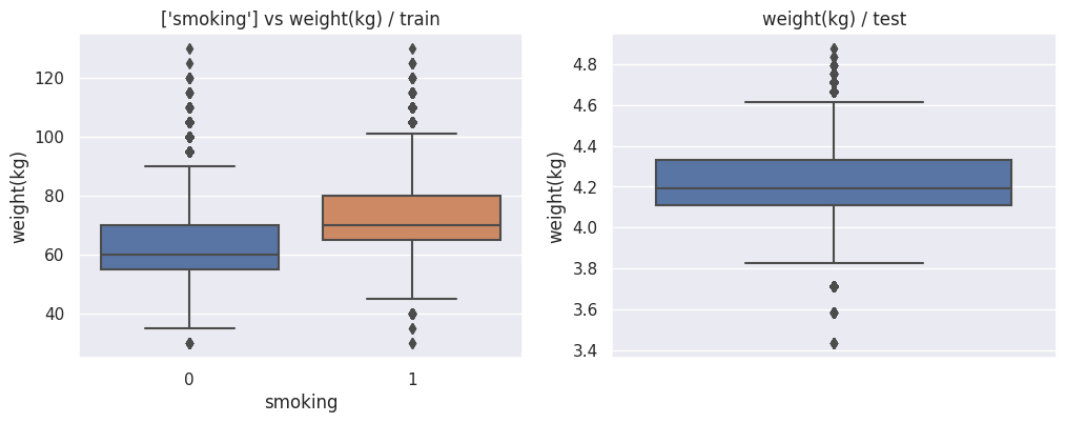


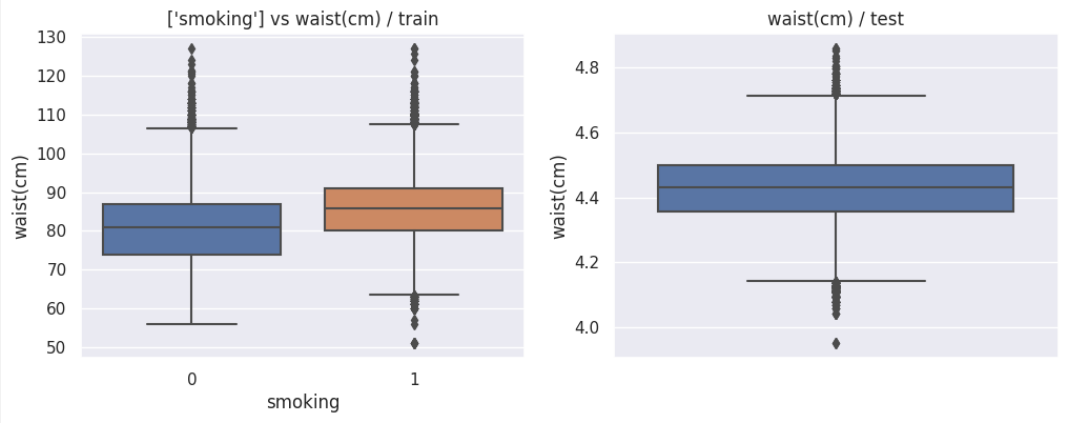

```
defects False 78699 True 23064
Name: count, dtype: int64
```

箱ひげ図

sns.boxplot()

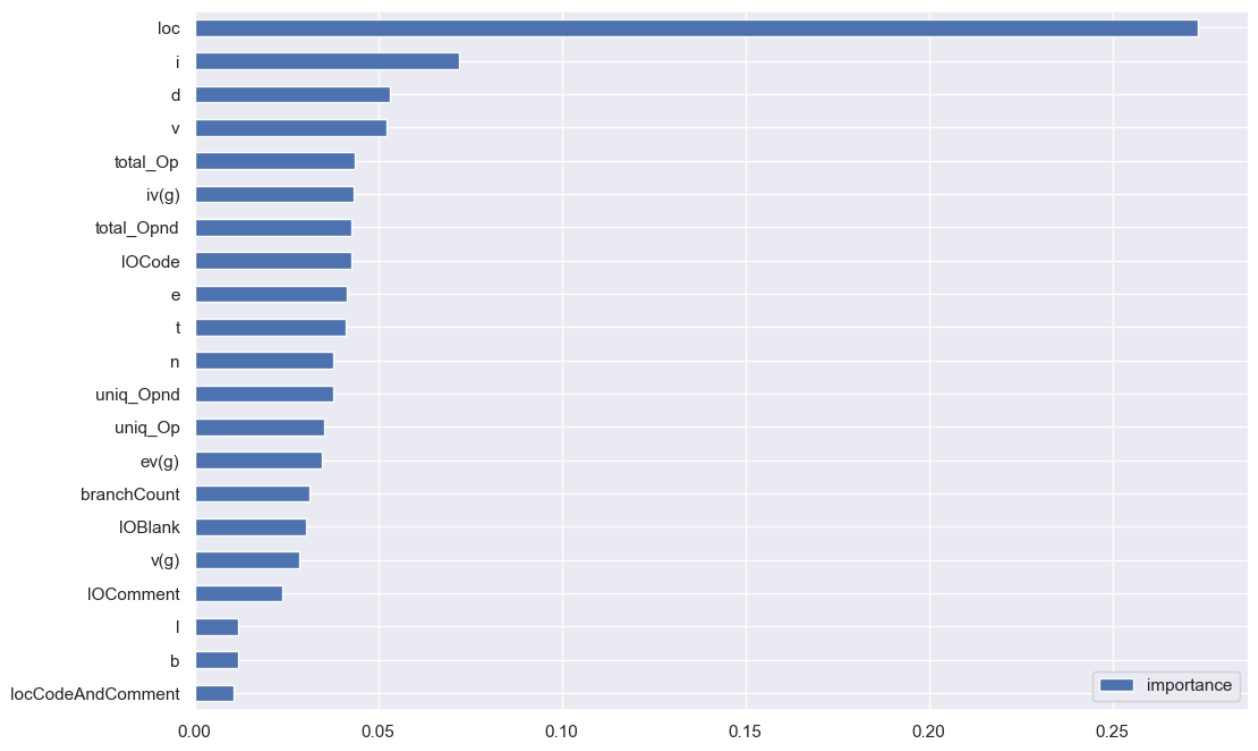






特徴量の重要度評価

RandomForestRegressor()

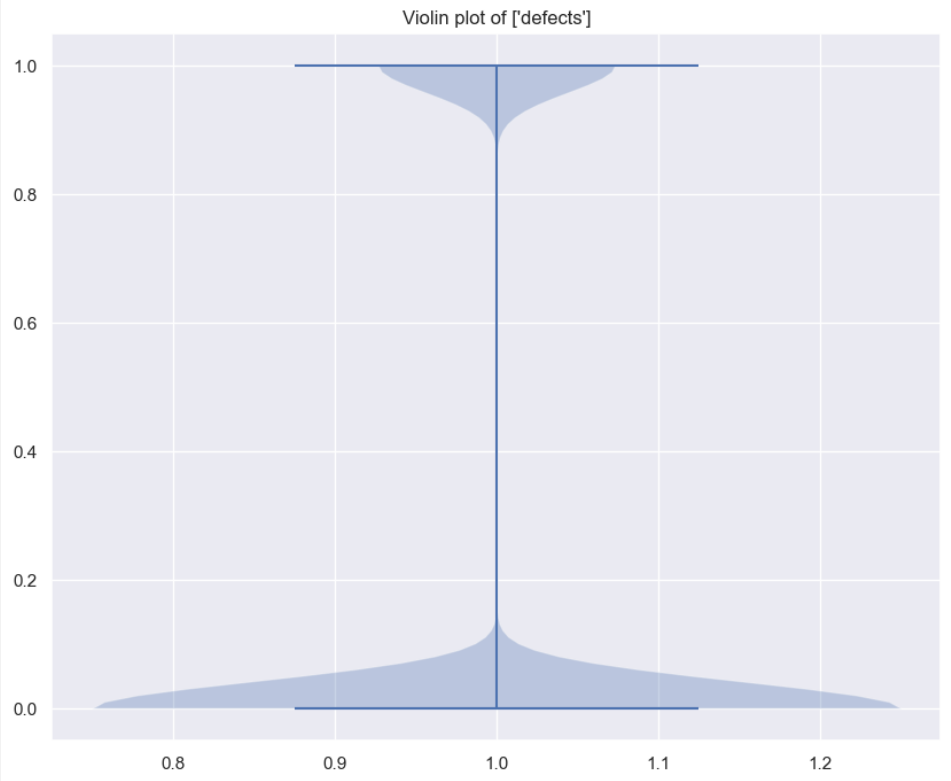



```
   importance
loc 0.273385
i 0.071989          uniq_Op 0.035412
d 0.053248          ev(g) 0.034578
v 0.052177          branchCount
total_Op 0.043724 0.031397
iv(g) 0.043218      lOBlank 0.030468
total_Opnd          v(g) 0.028589
0.042813            lOComment
lOCode 0.042784     0.023841
e 0.041320          l 0.011950
t 0.041061          b 0.011861
n 0.037793          locCodeAndComment
uniq_Opnd           0.010735
0.037657
```

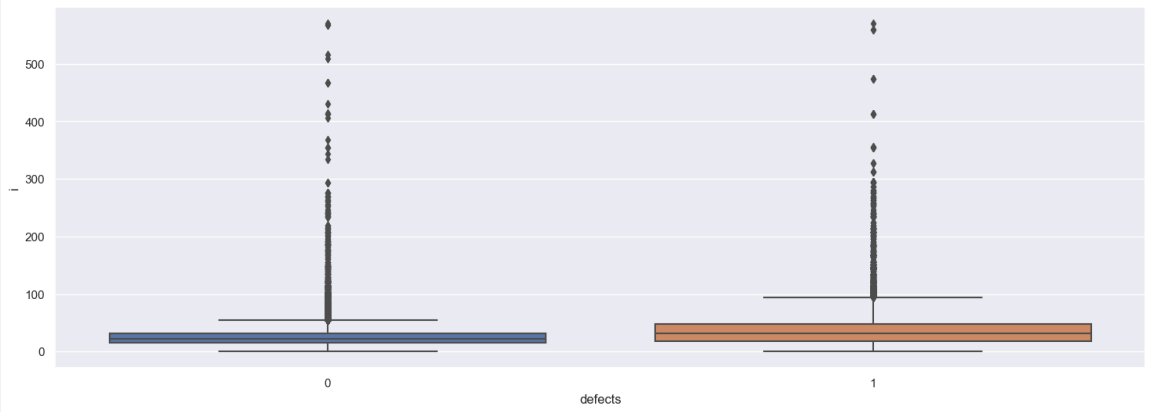
箱ひげ図

plt.boxplot(), plt.violinplot()

データの範囲を調べる

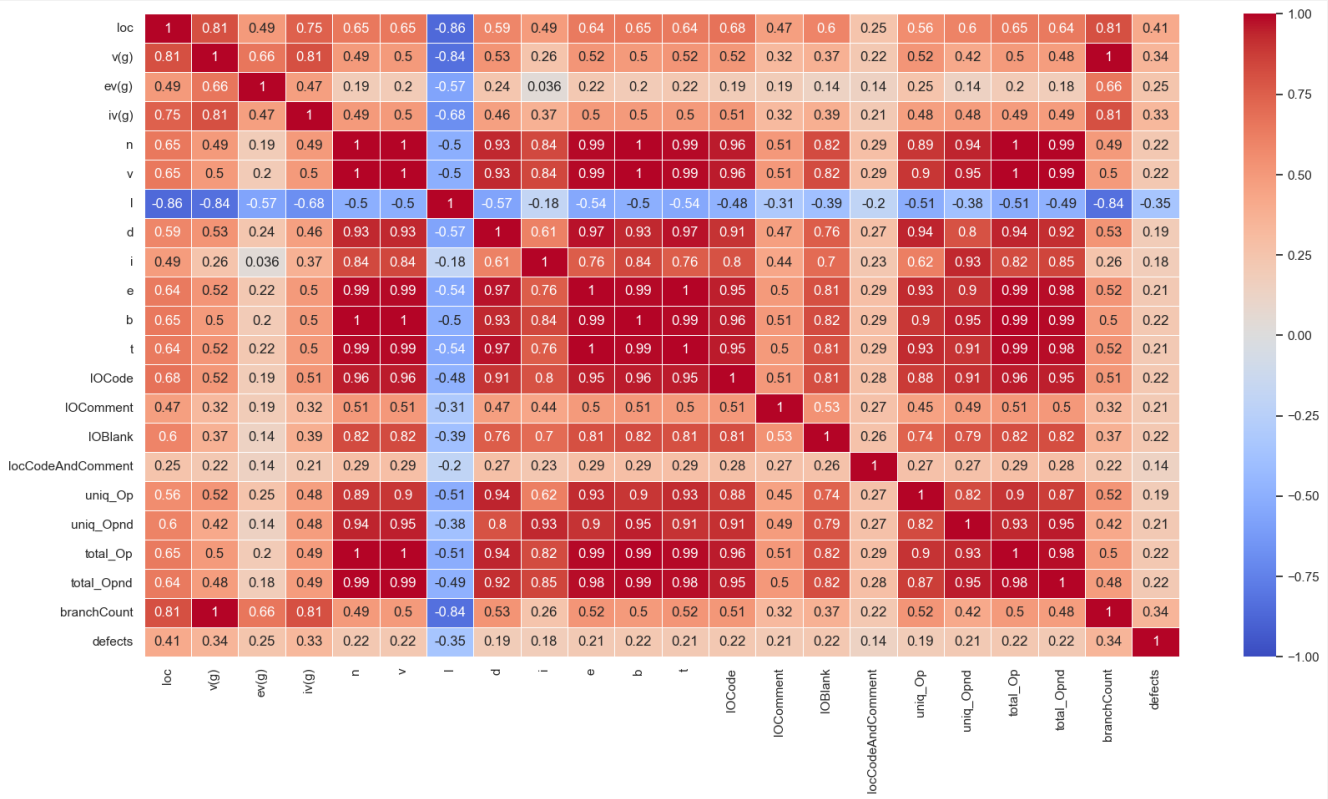


目的変数と'i'（最も特徴重要度の高い説明変数）との関係



相関（ヒートマップ）

sns.heatmap()

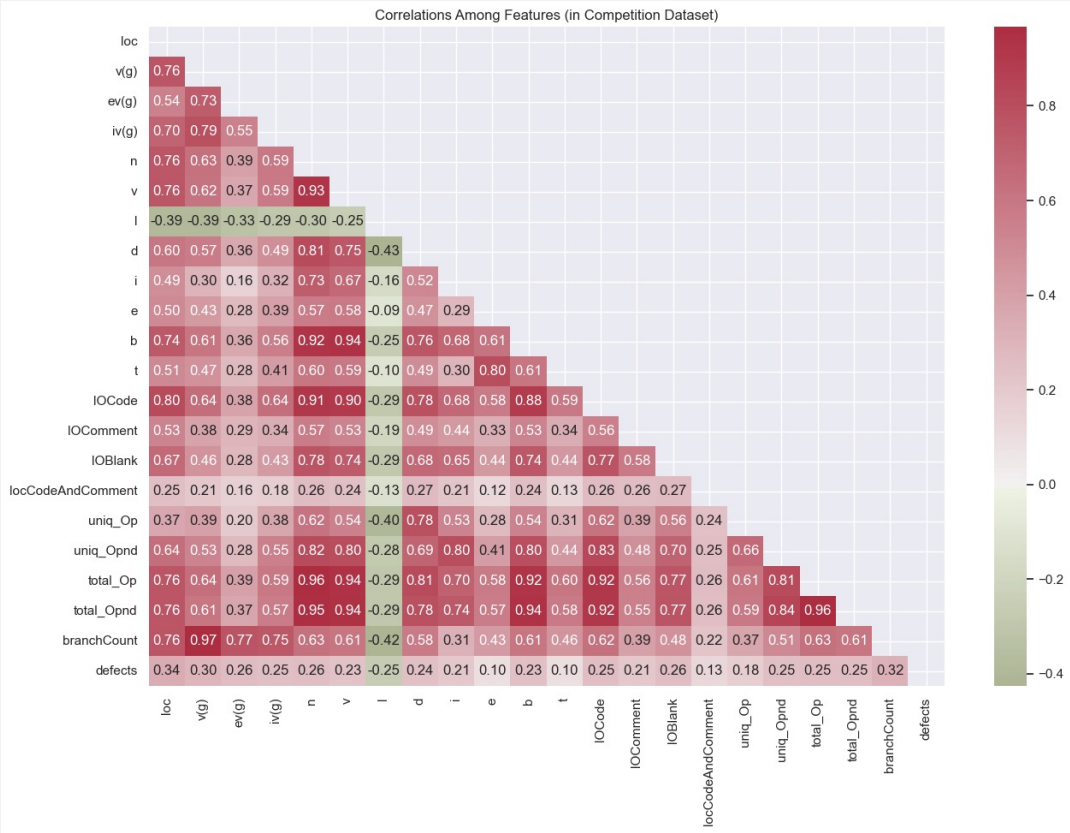


相関係数が1の組み合わせ

- 'branchCount', 'v(g)'
- 'n', 'v', 'b', 'total_Op'(bとは0.99)
- 'e', 't'

相関（目的変数）

sns. heatmap()



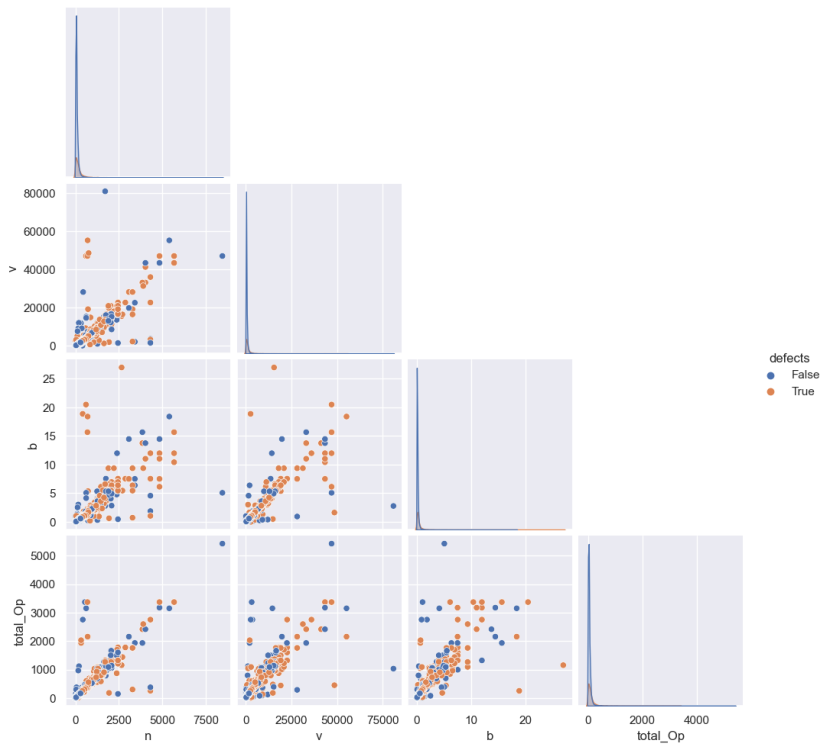
目的変数との相関係数

```
defects 1.000000
loc 0.342642        iv(g) 0.245618
branchCount         d 0.241936
0.322827            b 0.232594
v(g) 0.301187       v 0.231179
ev(g) 0.259928      i 0.208577
n 0.258080          lOComment
lOBlank 0.257819    0.205402
total_Opnd          uniq_Op 0.178474
0.252752            locCodeAndComment
lOCode 0.250604     0.133150
total_Op 0.250533   t 0.099592
uniq_Opnd           e 0.095366
0.246113            l -0.253237
```

相関（ペアプロット）

sns.pairplot()　sort_values()

'n', 'v', 'b', 'total_Op'と
defects(目的変数)との相関関係を調べる



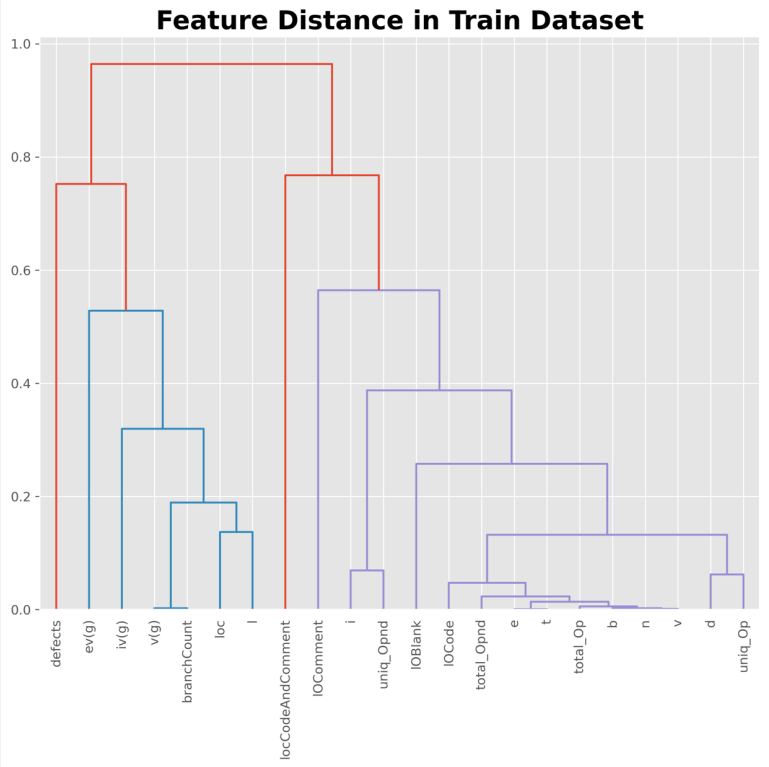
目的変数と最も相関係数が高い説明変数の値ごとの
データ数を数える

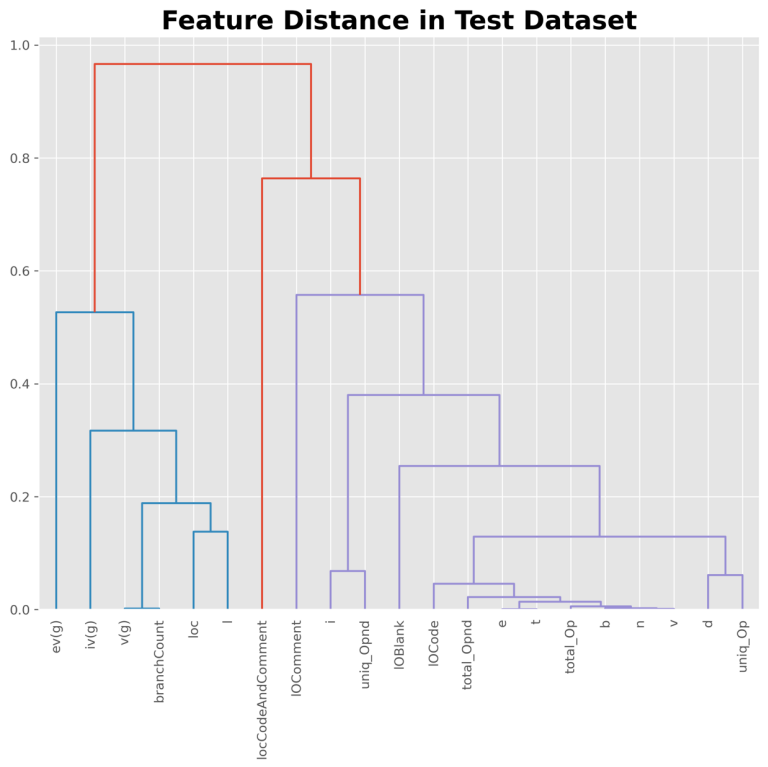
```
loc
11.0 3657
12.0 3431
7.0 3426
17.0 3255
8.0 3109
10.0 3084
14.0 3083
15.0 2935
9.0 2797
13.0 2713
dtype: int64
```

階層的クラスタリング

dendrogram()

- 特徴間の関係の強さと方向性を調べる
- 特徴をクラスタリングする





参考資料

書籍

- 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–タ分析
- Python実践データ加工:可視化100本ノック
- ［第3版］Python機械学習プログラミング 達人データサイエンティストによる理論と実践

Kaggle

- PSS3E23 EDA which makes sense ⭐⭐⭐⭐⭐
- PS_3.23_LGBM_bin_日本語あり
- PS3E23 | Binary Classification for Beginners

自分で作成したファイル

- モース硬度データセットを使用した回帰
- 生体信号を使用した喫煙者の状態のバイナリ予測