One Class SVMを用いた異常値検知

Anomaly detection using One Class SVM

公立はこだて未来大学　森雄斗

概要

- 教師なし学習により、データのパターンを学習させ、パターンから外れたデータを異常値として検出する
- 検出方法は、One Class SVMを利用する。

- 前回マハラノビス距離で行ったことを再度One Class SVMで行う。

→https://www.slideshare.net/YutoMori2/ss-88160534

One Class SVM とは

- SVM（サポートベクトルマシン）の中の一種。
- SVMは教師あり学習であるのに対して、OneClassSVMは教師なし学習である。
- One Class SVMは異常値検知によく用いられる。

実験環境

- Python 3.6.0

  numpy, pandas, mmatplotlib, sklearn, time, plotly

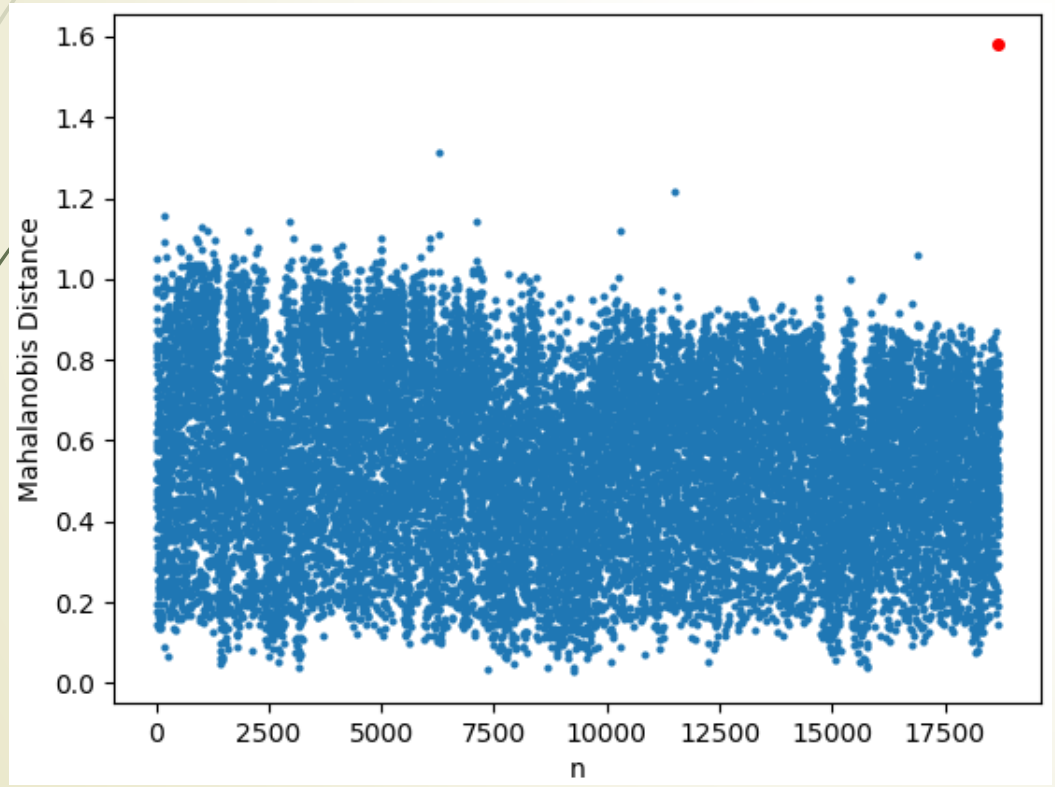
- 実際のデータ(全てが正常値)

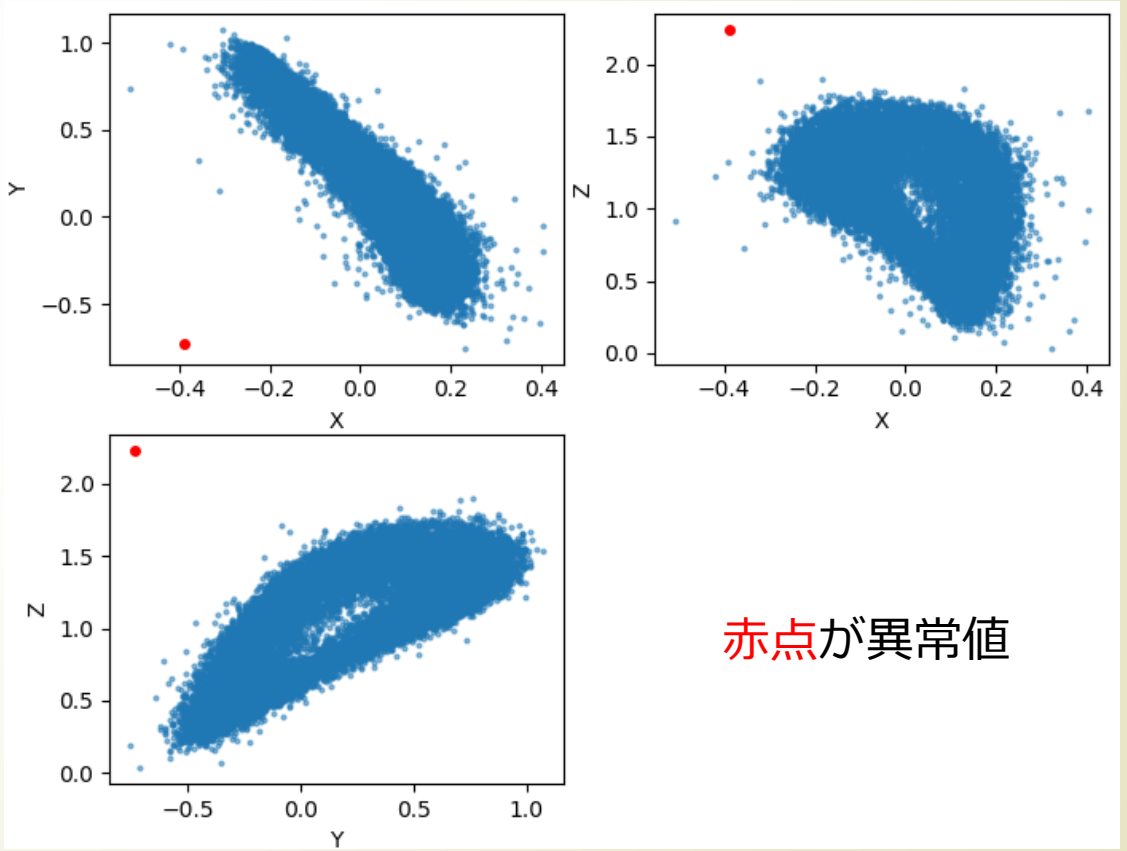
  MachineA.csv → 22670個 × 3 (x軸, y軸, z軸)

  MachineB.csv → 18700個 × 3 (x軸, y軸, z軸)

MachineBのデータに異常値を追加する

マハラノビス距離が1.5794である異常値(-0.39, -0.73, 2.235)を追加する。





Class sklearn.svm.OneClassSVM

One class SVMを適応するのは2行

```
clf = svm.OneClassSVM()
clf.fit(データ)
```

課題はパラメータの調整(チューニング)

パラメータについて

- デフォルトのパラメータ

OneClassSVM(cache_size=200, coef0=0.0, degree=3, gamma='auto', kernel='rbf', max_iter=-1, nu=0.5, random_state=None, shrinking=True, tol=0.001, verbose=False)

- 今回重要なのは"gamma"と"nu".
- gamma = 'auto' = 1 / n_features = 0.5

つまり、3変数でも出来るのでは....

パラメータについて

- nu

異常データの割合….？（0~1）

- gamma

RBFカーネルのパラメータ…？

→値が大きいほど境界が複雑になる…？

正直あまりわかっていないので実際に数値を変えて検証する

2変数分析

(XYの関係, XZの関係, YZの関係)

2変数分析の説明

- それぞれに対してSVMを用いる

```
dataXY = np.vstack([dataX, dataY]).T
dataXZ = np.vstack([dataX, dataZ]).T
dataYZ = np.vstack([dataY, dataZ]).T
```

```
clf.fit(dataXY)
y_pred = clf.predict(dataXY)
```

```
clf.fit(dataXZ)
y_pred = clf.predict(dataXZ)
```

```
clf.fit(dataYZ)
y_pred = clf.predict(dataYZ)
```

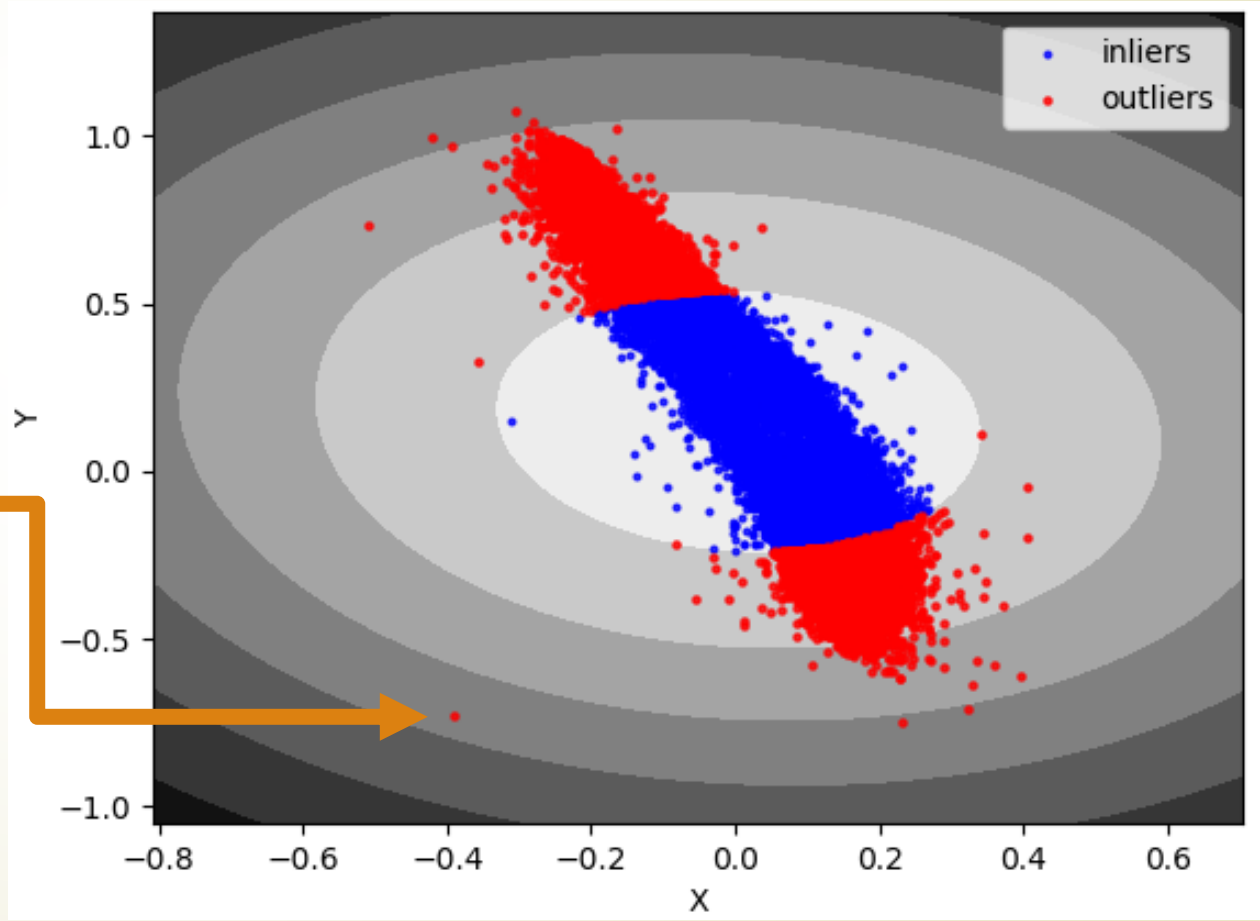
グラフの説明

- inliers → 正常値
- outliers → 異常値

- スライド5で追加したダミーの異常値

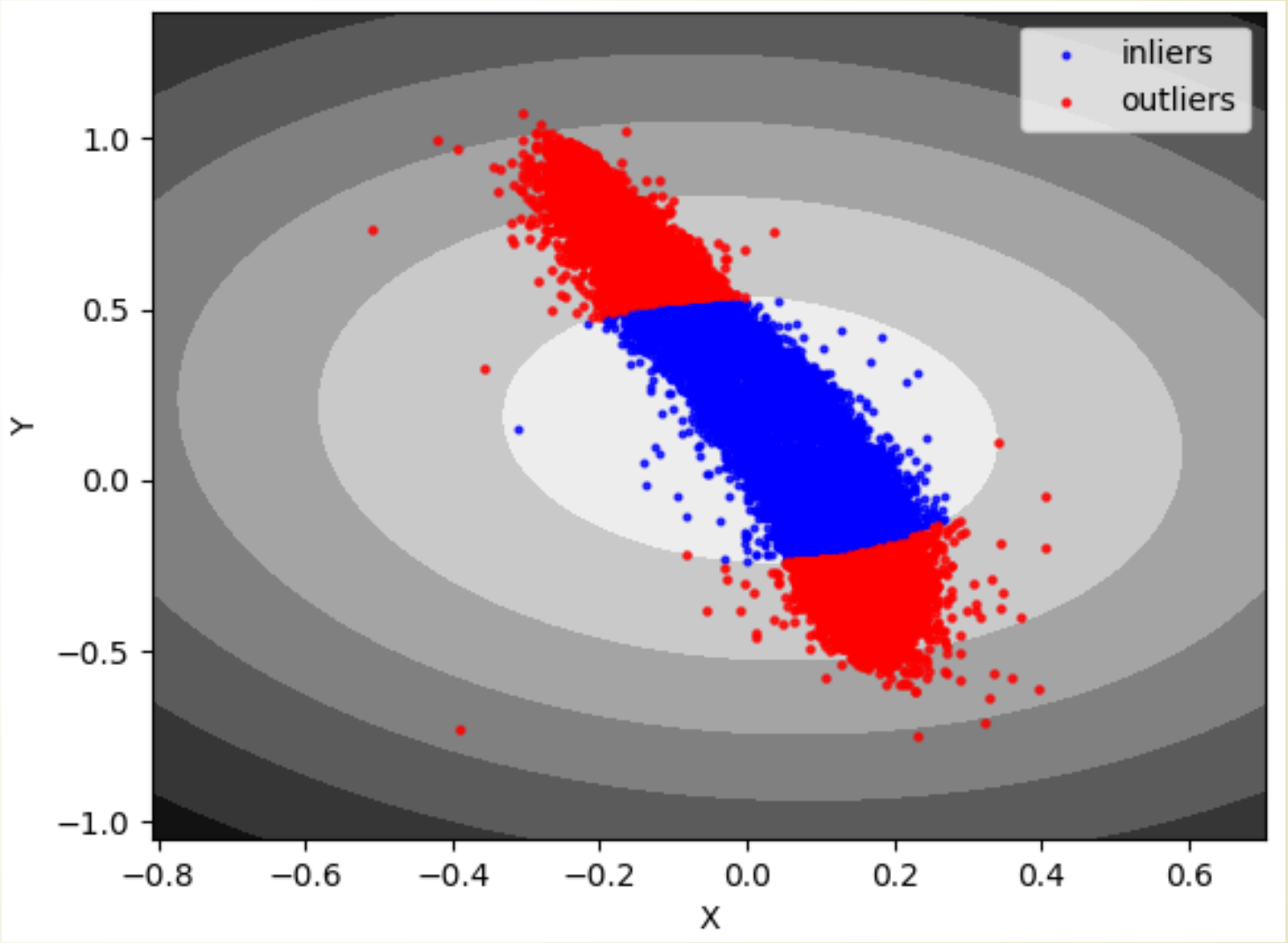
  (-0.39, -0.73, 2.235)

- グレーの等高線は異常度示している（黒になるほど異常値であると予測させる）



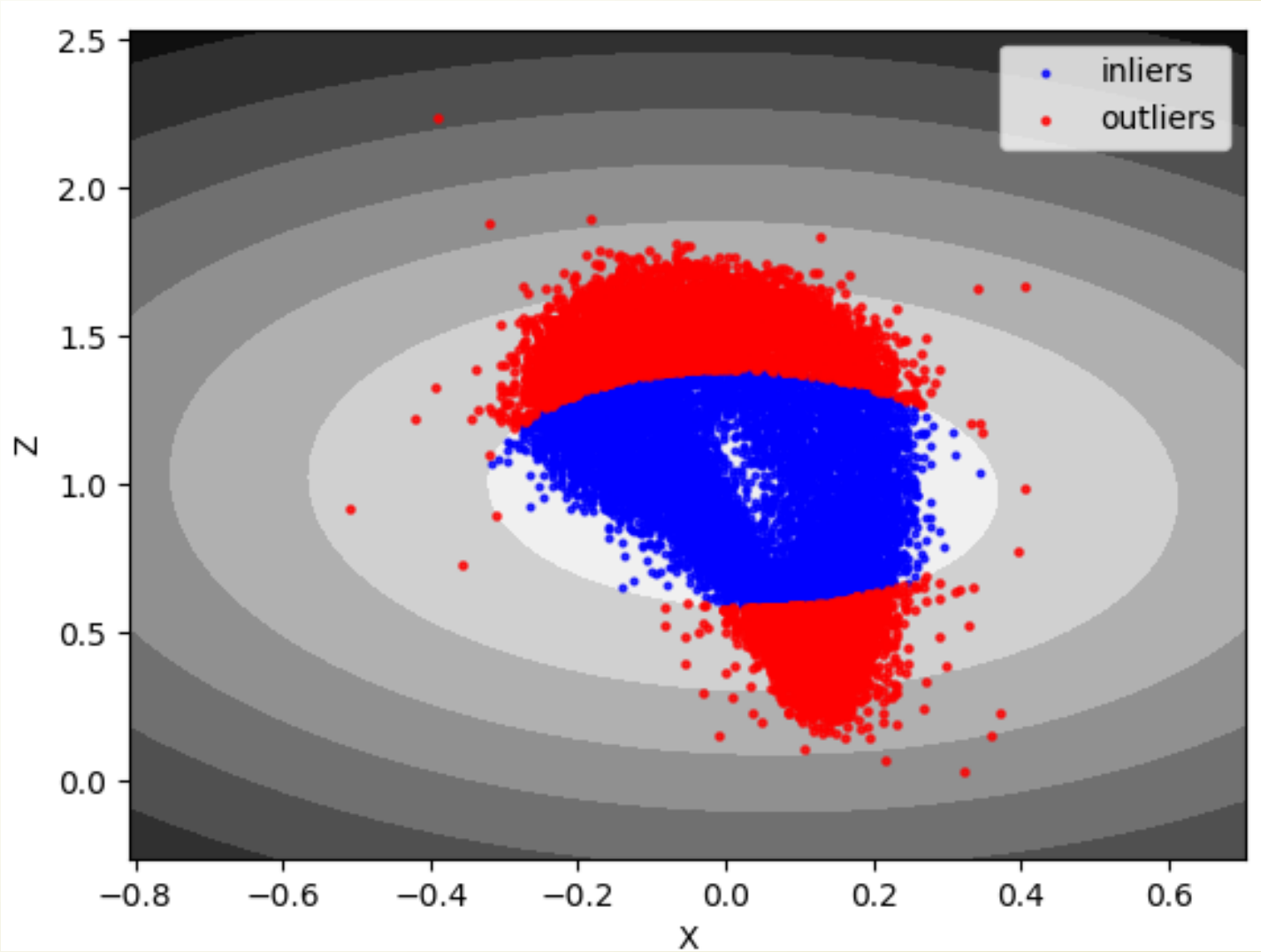
デフォルト（nu=0.5, gamma=0.5）

- XYの関係
- MachineB
- 実行時間 =194.1s



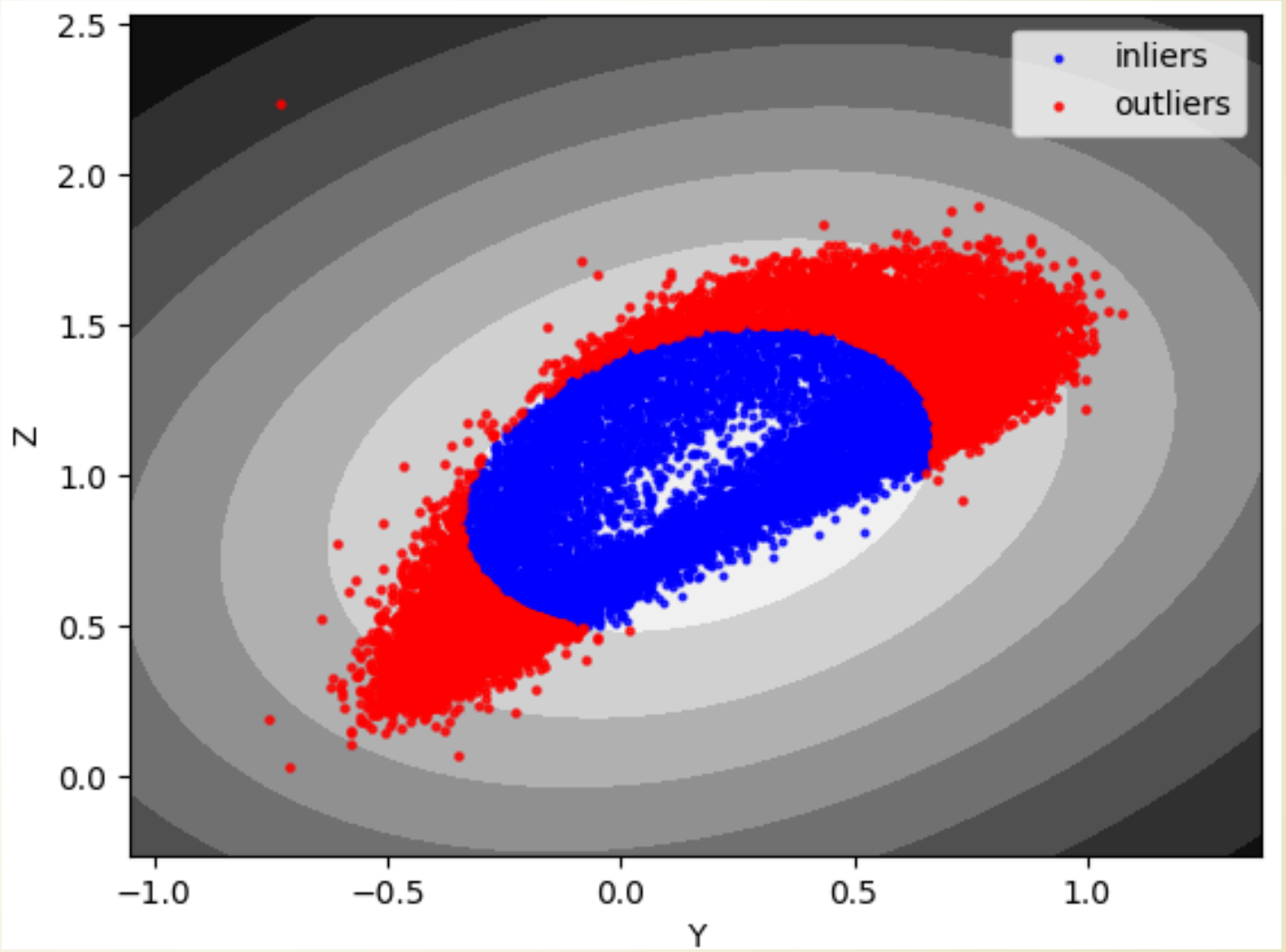
デフォルト（nu=0.5, gamma=0.5）

- XZの関係
- MachineB
- 実行時間 =194.1s



デフォルト（nu=0.5, gamma=0.5）

- YZの関係
- MachineB
- 実行時間 =194.1s

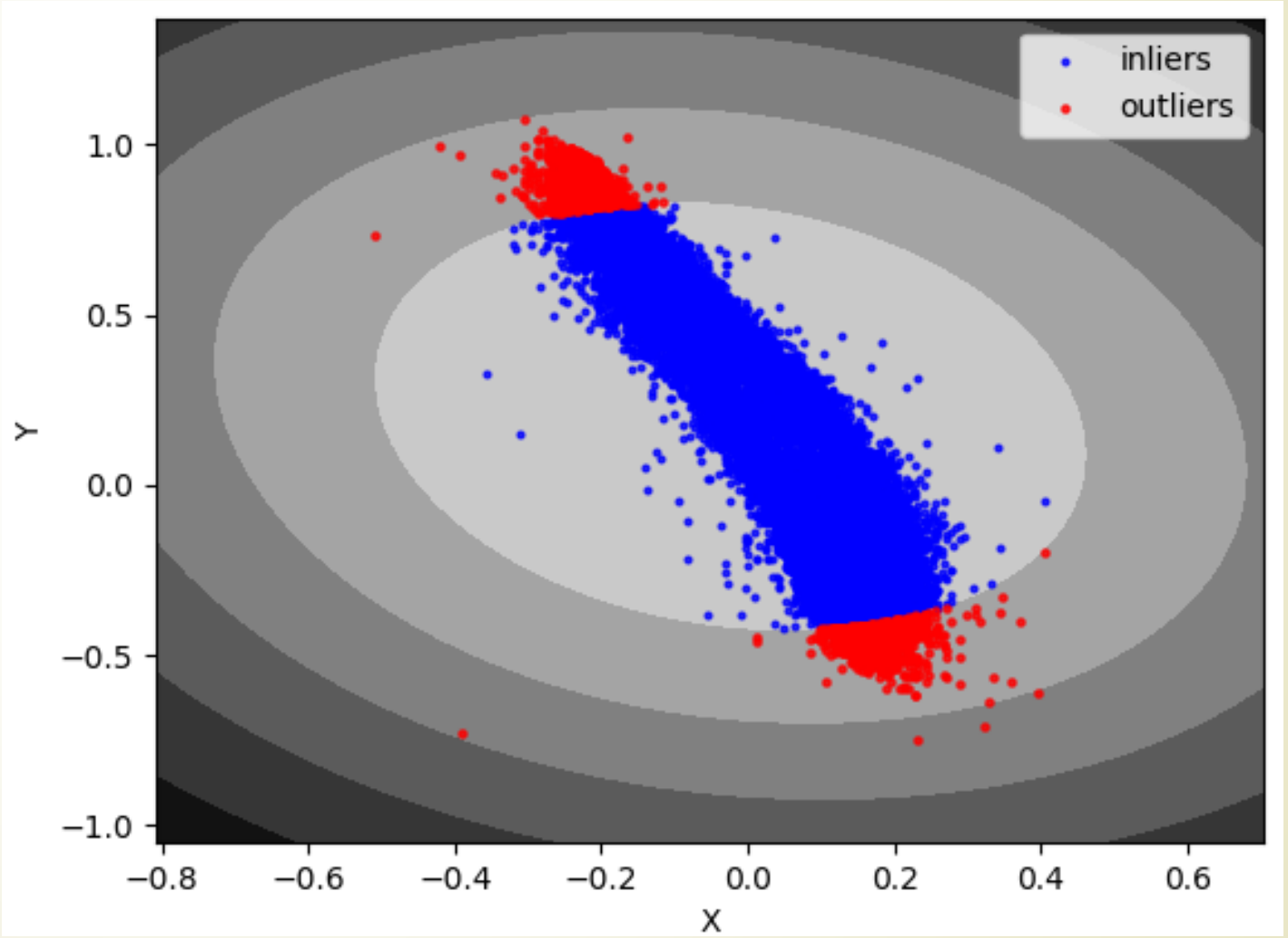


パラメータ調整

nu = 0.5　→　nu = 0.1

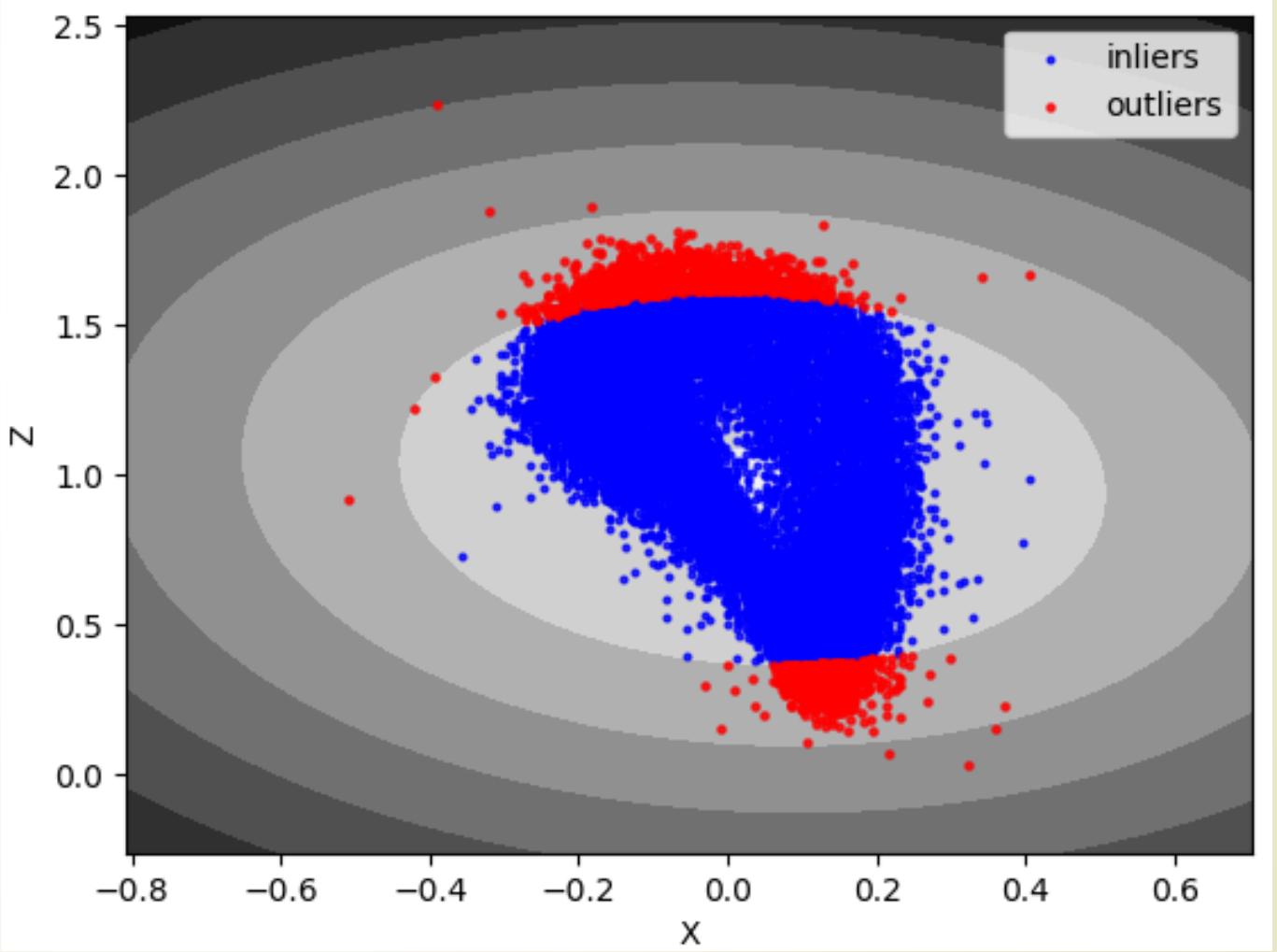
nu=0.1, gamma=0.5

- XYの関係
- MachineB
- 実行時間 =37.2s



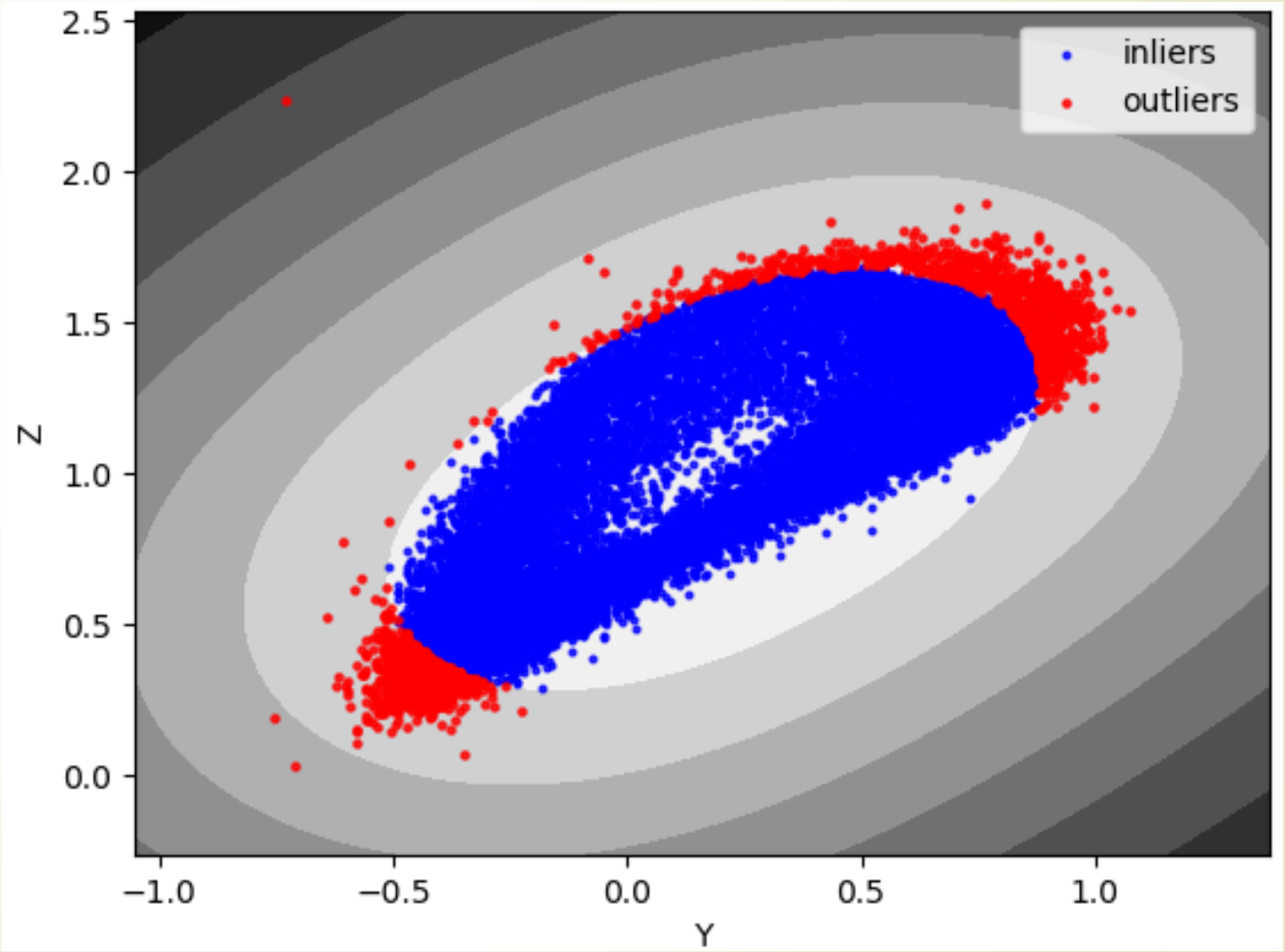
nu=0.1, gamma=0.5

- XZの関係
- MachineB
- 実行時間 =37.2s



nu=0.1, gamma=0.5

- YZの関係
- MachineB
- 実行時間 =37.2s



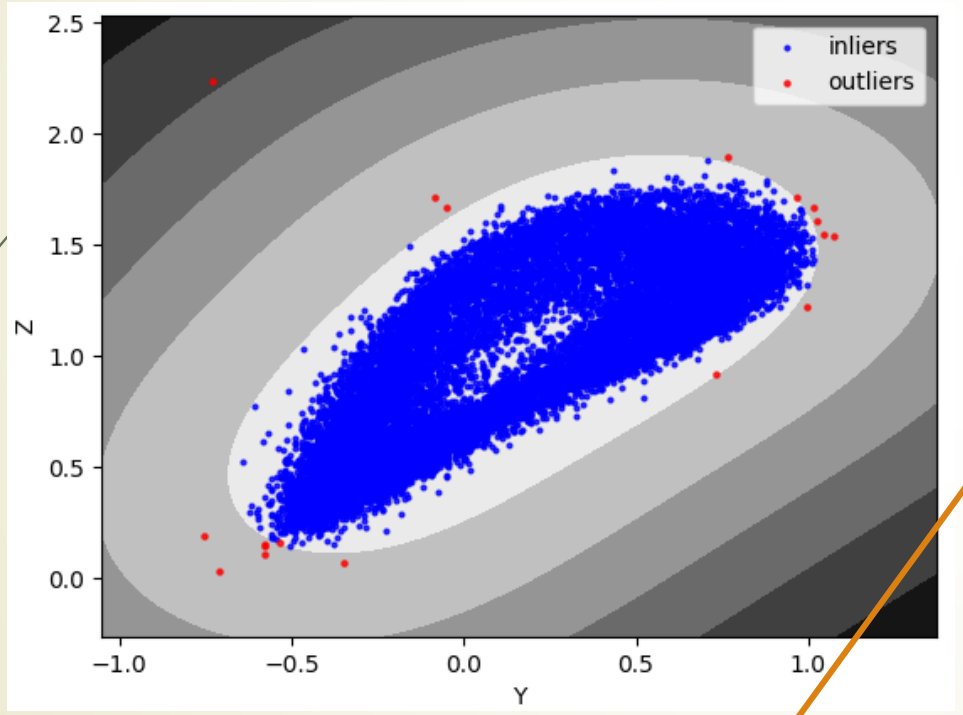
パラメータ調整

この作業を繰り返す….

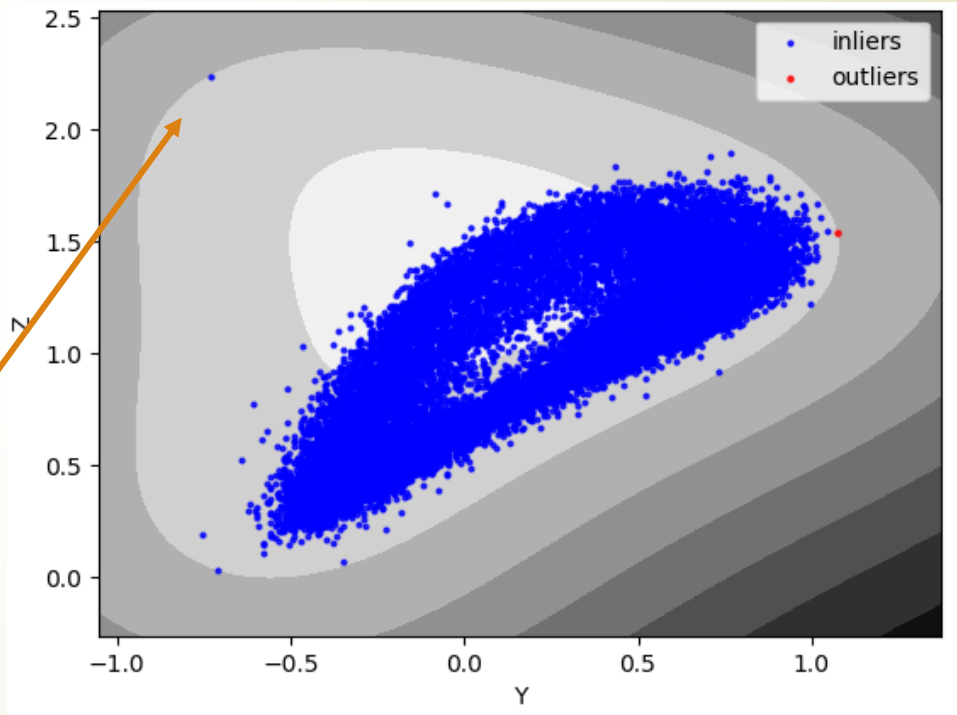
パラメータ nu について

gamma = 0.5を固定
XYの関係

nu = 0.001



nu = 0.0001



0.001~0.0001の間に異常値を正常値と判断し、異常度等高線が大きく変化する

パラメータ nu について

○入力データ

MachineB_dummy.csv
正常値: 18700, 異常値: 1, 計: 18701

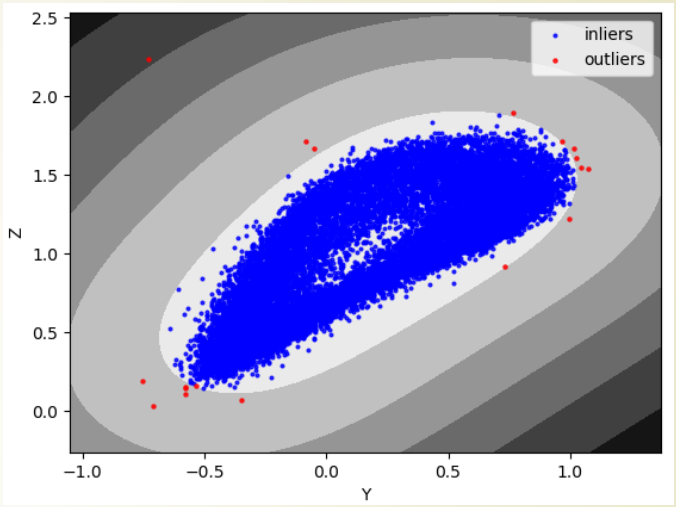
●nu = 0.001のOne Class SVMを適応したグラフ(1)
正常値: 18682, 異常値: 19, 計: 18701

18701 * 0.001 = 18.7 ≒ 19

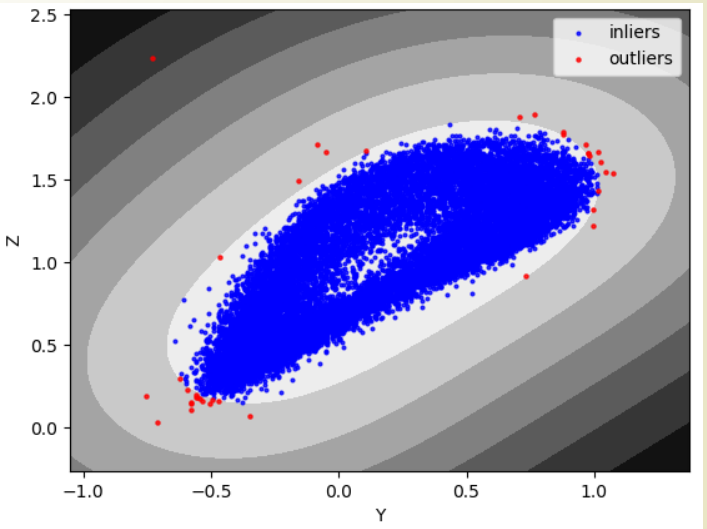
●nu = 0.002のOne Class SVMを適応したグラフ(2)
正常値: 18664, 異常値: 37, 計: 18701

18704 * 0.002 = 37.4 ≒ 37

(1) nu = 0.001



(2) nu = 0.002



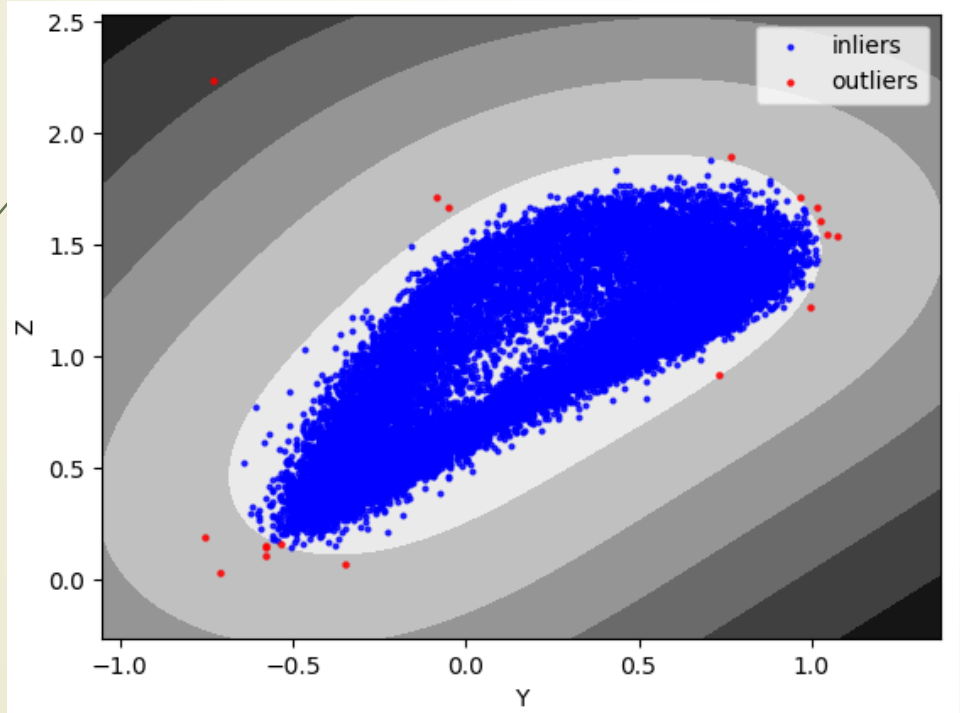
パラメータ nu のまとめ

- パラメータ nu は入力データの異常値の割合であることが明らかである。
- 異常値の割合が0.0001のような小さい値をとると極度に異常であるデータを正常値と判断してしまう可能性がある。
- nuが小さくなればなるほど、異常度等高線が適切なものになる。

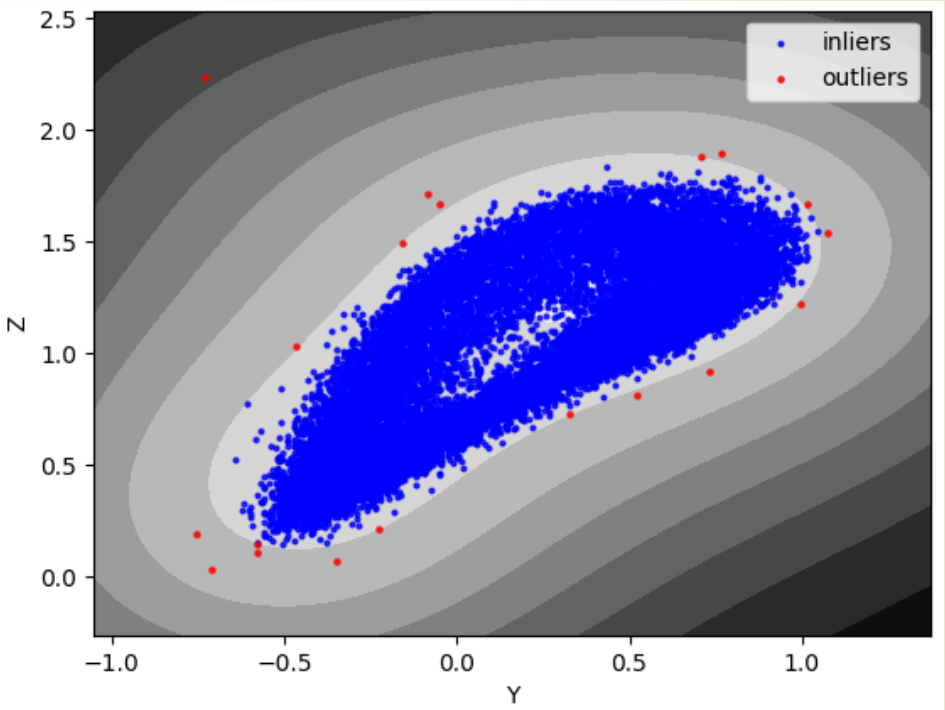
パラメータ gamma について

nu = 0.001を固定
YZの関係

gamma = 0.5



gamma = 1.0

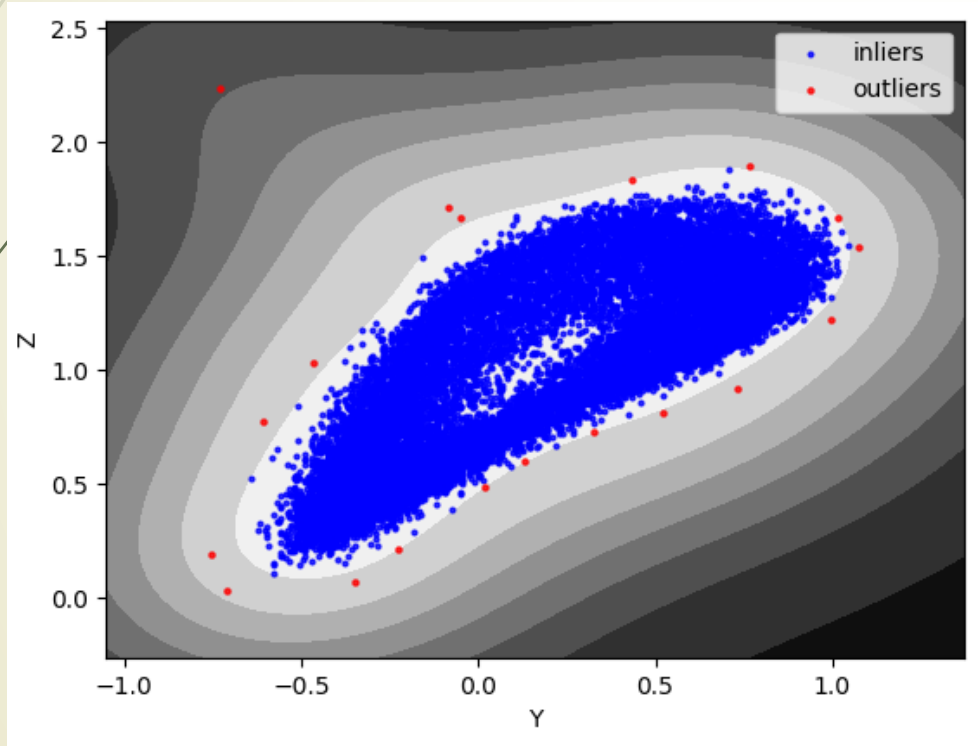


gamma値の大小で汎化能力が変化する
(汎化能力: 規則性の当てはまり)

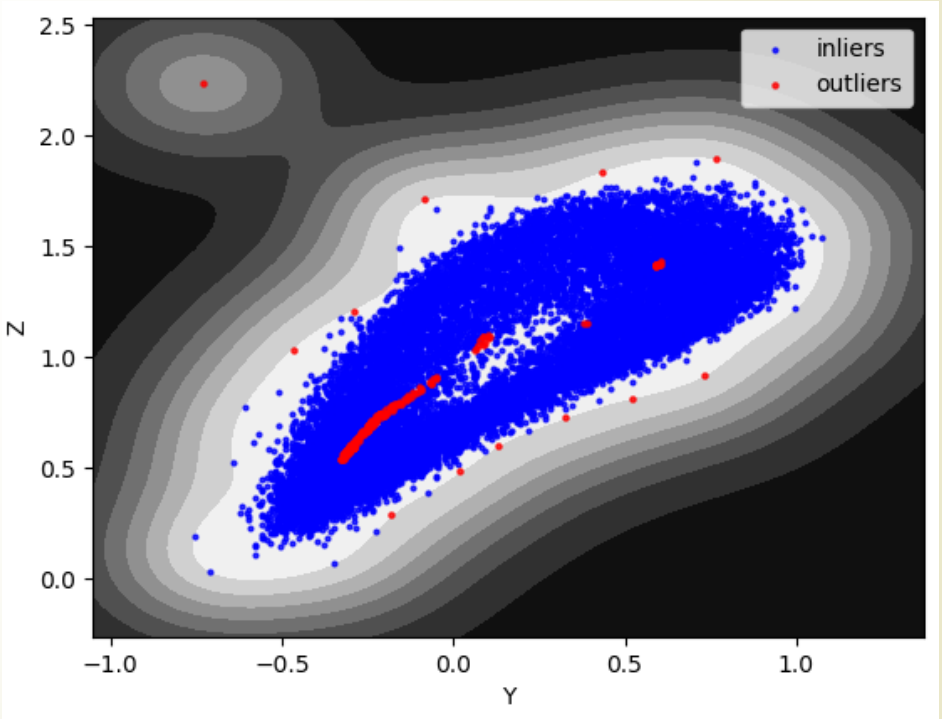
パラメータ gamma について

nu = 0.001を固定
YZの関係

gamma = 3.0

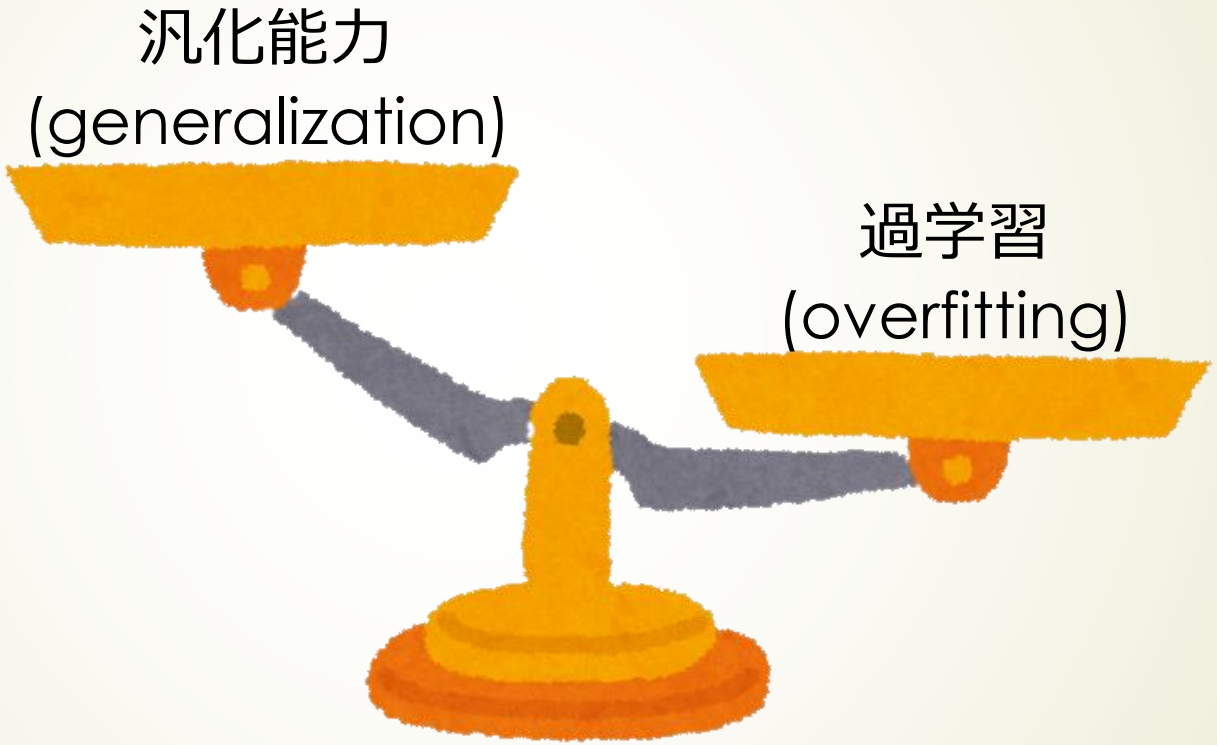


gamma = 10.0



gamma値を上げすぎると過学習状態に…

汎化能力と過学習



最適な学習モデルは、「過学習が起こらず、汎化性能に優れたもの」

3変数分析

(XYZの関係)

3変数分析の説明

python

```
dataXYZ = np.vstack([dataX, dataY, dataZ]).T
```

```
clf.fit(dataXYZ)
y_pred = clf.predict(dataXYZ)
```

print

dataXYZ

```
[[ 0.068  0.228  1.484]
 [-0.18   0.816  1.344]
 [-0.108  0.392  0.932]
 ...,
 [-0.132  0.692  1.572]
 [-0.168  0.564  1.076]
 [-0.39  -0.73   2.235]]
```

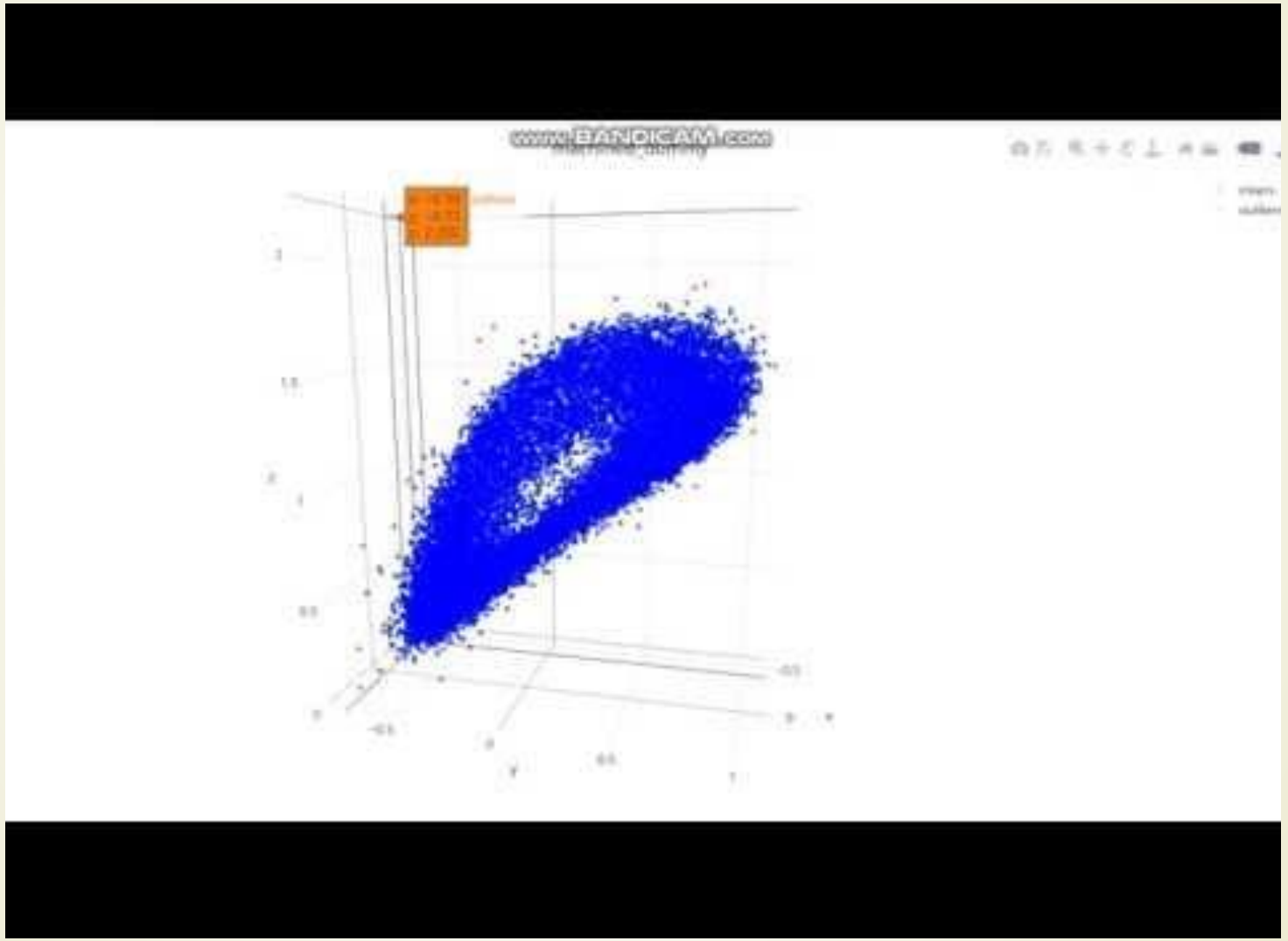
y_pred

```
[ 1  1  1 ...,  1  1 -1]
```

1: 正常値
-1: 異常値

plotlyによる描写

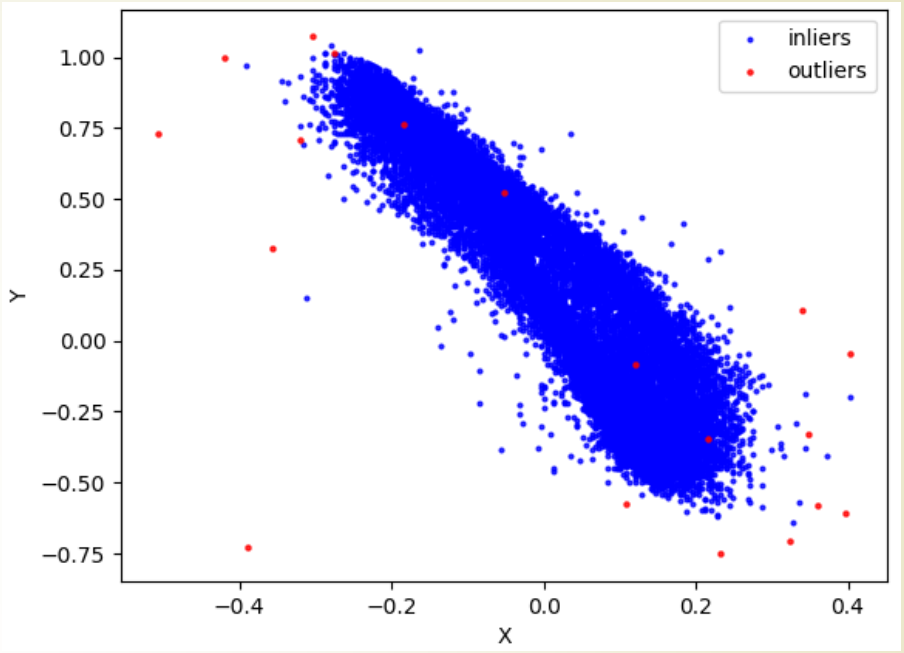
パラメータ:
Nu = 0.001, gamma = 1

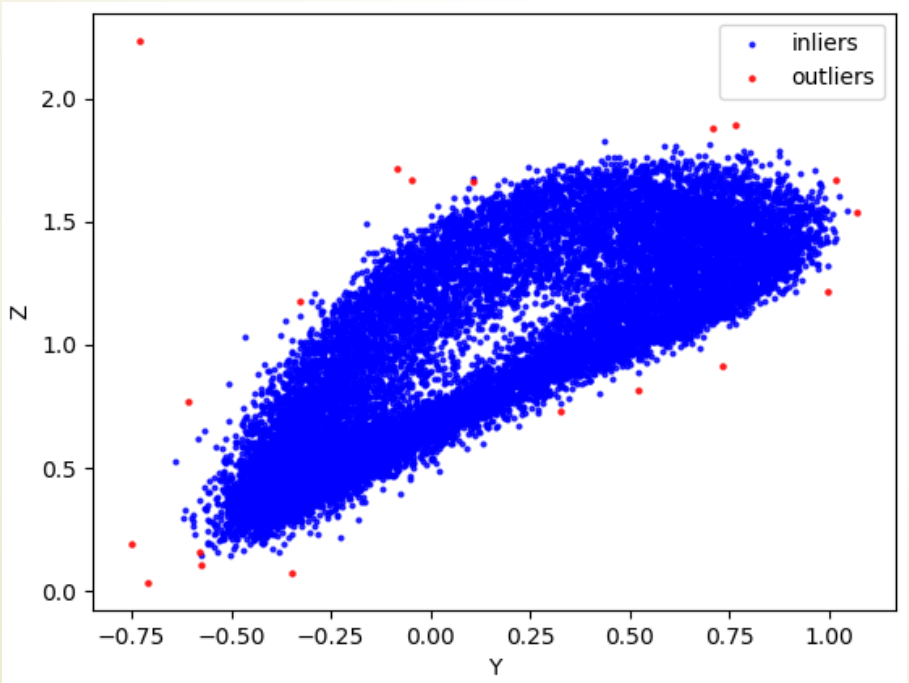


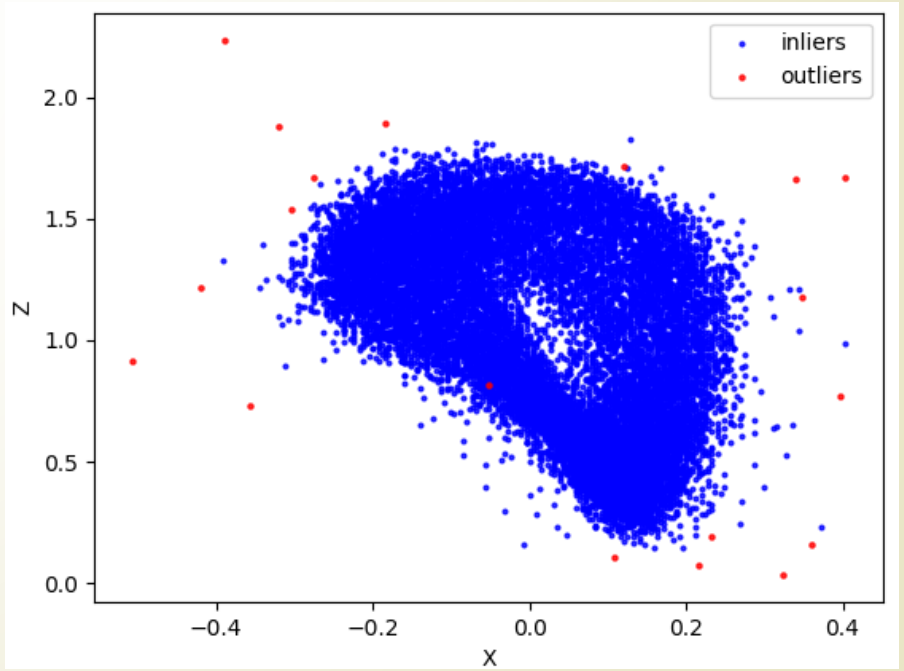
URL：https://www.youtube.com/watch?v=Tu3wEj0Inc0

matplotlibによる描写

3変数で分析したものを
2変数に変換したもの



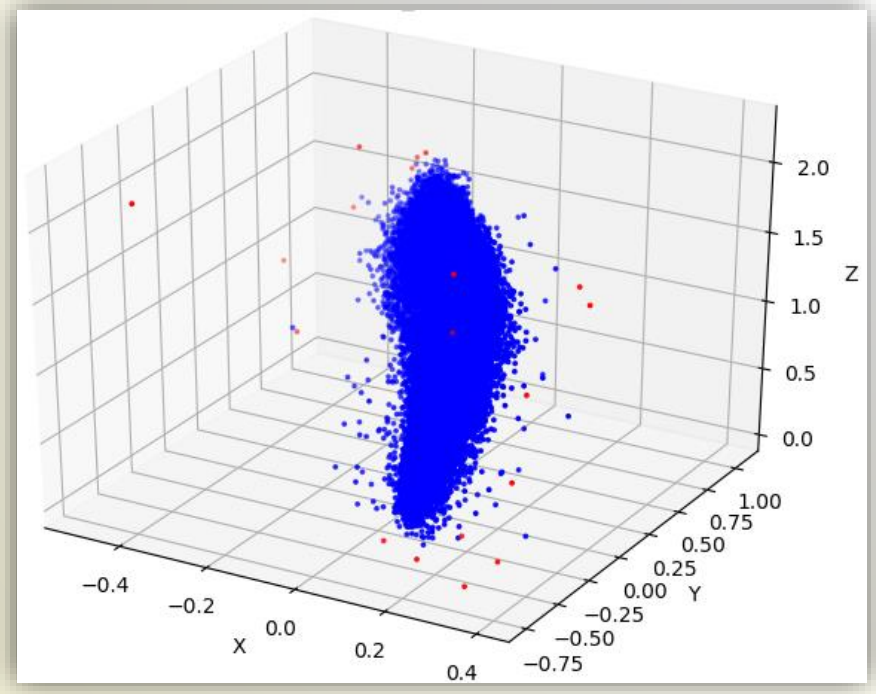




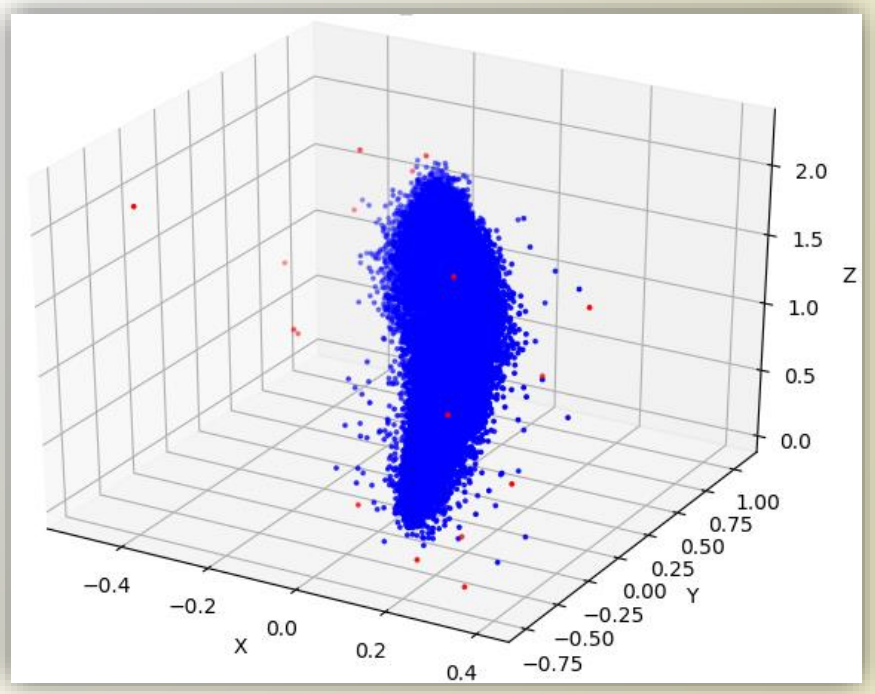
3変数分析の問題点

- 異常度等高線を描写できない。
- グラフを見ても適切なパラメータに設定することができない。

nu = 0.001, gamma = 1.0



nu = 0.001, gamma = 2.0



分類器の性能評価（分割表）

予測

実際	分割表	正常値(inliers)	異常値(outliers)
	正常値(inliers)	True Positive(TP)	False Negative(FN)
	異常値(outliers)	False Positive(FP)	True Negative(TN)

分類器の性能評価(Weighted F-measure)

- 重み付きF値(Weighted F-measure) とは
  - ラベル間のデータ数が大きく異なる場合によく使われる指標
  - 今回は偽陽性(本当は異常値であるのに検査結果で正常と出ること)を低く抑えたいので, 適合率に重さを付ける(1 < β)

- 適合率(Precision) $= \frac{True\ Positive}{True\ Positive + false\ Positive}$
- 再現率(Recall) $= \frac{True\ Positive}{True\ Positive + False\ Negative}$

- 重み付きF値(Weighted F-measure) $= \frac{(1+\beta^2)\cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}$

https://www.quora.com/What-is-meant-by-F-measure-Weighted-F-Measure-and-Average-F-Measure-in-NLP-Evaluation

偽陽性率と真陽性率

- False Positive Rate(偽陽性率) = $\frac{False\ Positive}{False\ Positive + True\ Negative}$
  - 異常値であるものを間違って正常と予測した割合
- True Positive Rate(真陽性率) = $\frac{True\ Positive}{True\ Positive + False\ Negative}$
  - 正常値であるものを正しく正常と予測した割合

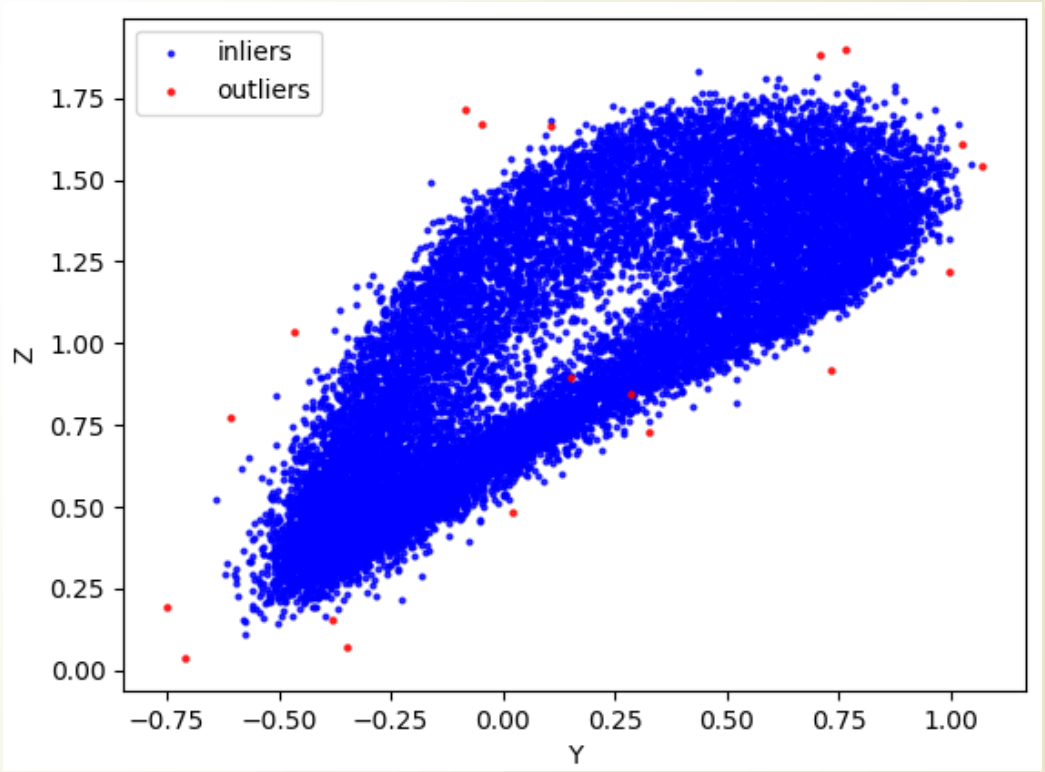
3変数分析のダミーを設定(Y軸, Z軸のみ)

- 右図に該当しない候補

(x, y, z)

= (?, -0.75, 1.9), (?, 0.00, 2.0),
(?, 0.65, 2.1), (?, 1.7, 1.00),
(?, 1.88, 0.65), (?, 0.75, 0.2),
(?, -1.2, 0.25), (?, -0.50, -1.2),

実際のデータのみ



3変数分析のダミーを設定(X軸, y軸, Z軸)

- 右図に該当しないように設定

(x, y, z)=

(-0.45, -0.75, 1.9), (0.9, -0.75, 1.9),

(-0.7, 0.00, 2.0), (0.81, 0.00, 2.0),

(-0.8, 0.65, 2.1), (0.45, 0.65, 2.1),

(-0.3, 1.7, 1.00), (0.02, 1.7, 1.00),

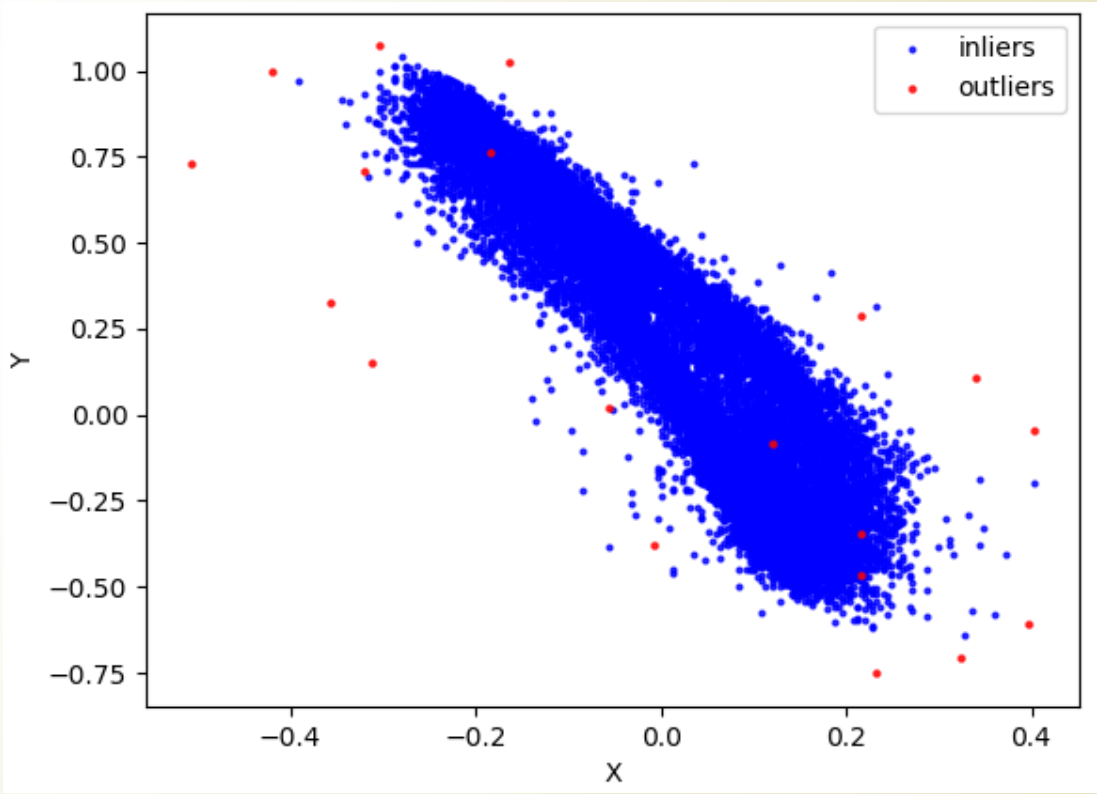
(-0.4, 1.88, 0.65), (-0.22, 1.88, 0.65),

(-0.91, 0.75, 0.2), (0.53, 0.75, 0.2),

(-0.21, -1.2, 0.25), (0.54, -1.2, 0.25),

(-0.71, -0.50, -1.2), (0.87, -0.50, -1.2)

実際のデータのみ



性能評価によるパラメータ設定

- 環境
  - データの数: 18716, nu = 0.00085 → 約16個選ばれる
  - 今回入れた異常値は16個

性能評価によるパラメータ設定

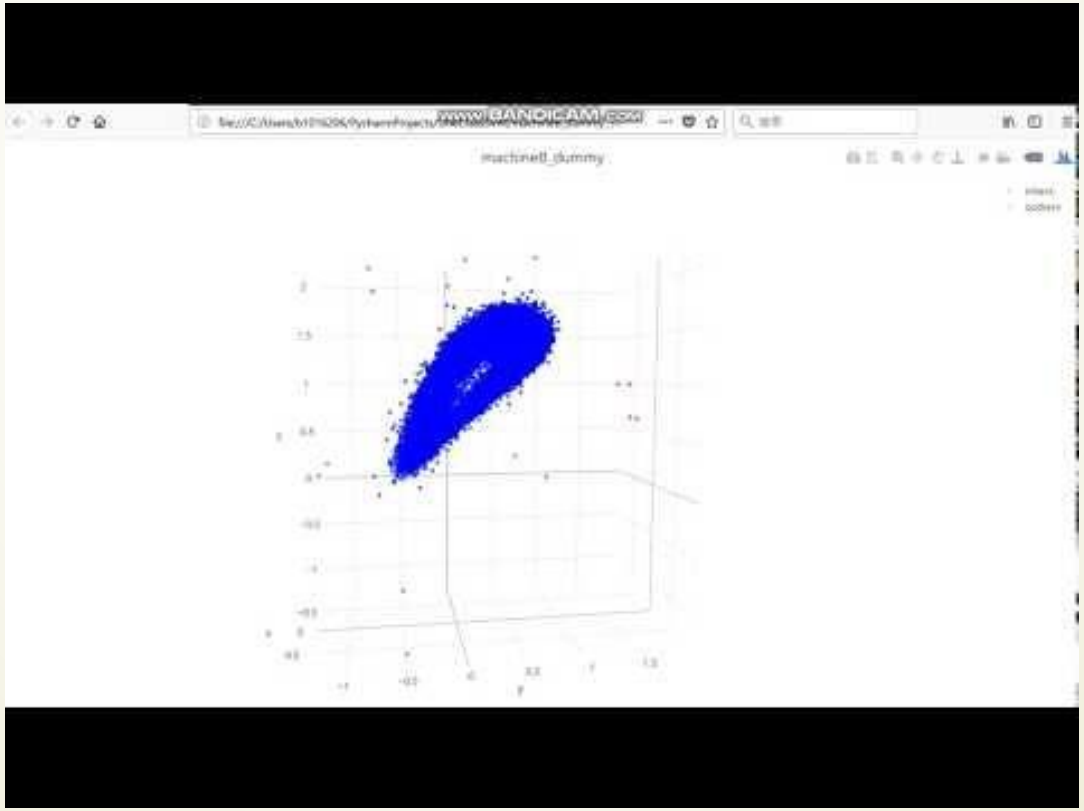
gamma	0.00001	0.00005	0.0001	0.001	0.01	0.1
True Positive	3690	18699	18698	18698	18699	18699
False Negative	15010	1	2	2	1	1
False Positive	0	4	1	2	1	2
True Negative	16	12	15	14	15	14
False Positive Rate(%)	0	25	6.25	12.5	6.25	12.5
True Positive Rate(%)	19.7326	99.973	99.989	99.9893	99.994	99.99465

性能評価によるパラメータ設定

gamma	0.33(auto)	0.5	1.0	2.0	3.5	5.0
True Positive	18699	18700	18697	18693	18689	18683
False Negative	1	0	3	7	11	17
False Positive	1	2	2	6	9	6
True Negative	15	14	14	10	7	10
False Positive Rate(%)	6.25	12.5	12.5	37.5	56.25	37.5
True Positive Rate(%)	99.99465	100.0	99.9839	99.9626	99.9412	99.9091

性能評価によるパラメータ設定

- 0.01 < gamma < 0.5 が適切なパラメータ
  - 右図が全ての異常値を正しく判断したgamma値(0.37)



URL : https://www.youtube.com/watch?v=6oACtaVSZ2A

gamma	0.37
True Positive	18700
False Negative	0
False Positive	0
True Negative	16
False Positive Rate(%)	0.0
True Positive Rate(%)	100.0

まとめ

- One Class SVM で教師なし学習を行った。
- 2変数分析と3変数分析に分けて機械学習を行った。
  - 2変数分析ではパラメータの特徴を理解し、3変数分析ではより正確な異常値検知を行うことができた。
- 性能評価を行い、適切なパラメータを見つけた。