Unsupervised Learning

Anomaly Detection

Peerapon Siripongwutikorn

Topics

Basic concepts in anomaly detection

Statistical-based techniques

- Mahalanobis distance
- Histogram based outlier scores

ML-based techniques

- Isolation forest
- One-class SVM

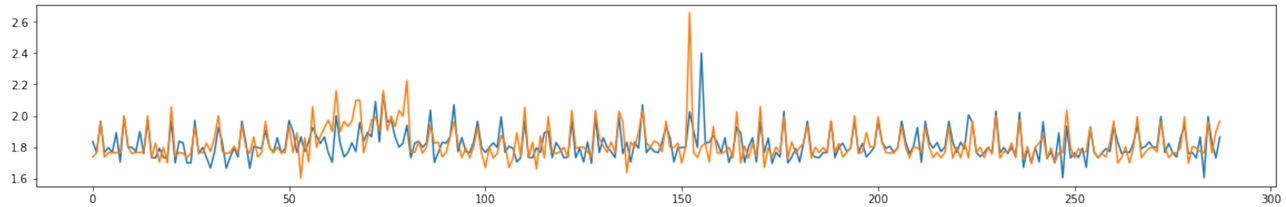
Latent-variable approach

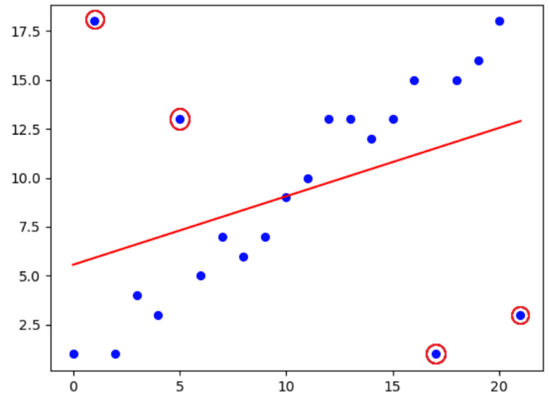
Anomalies

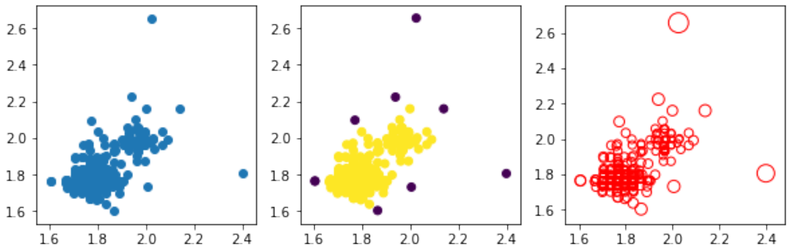
Somethings that deviate significantly from the majority (***inliers***) or different from expected behavior

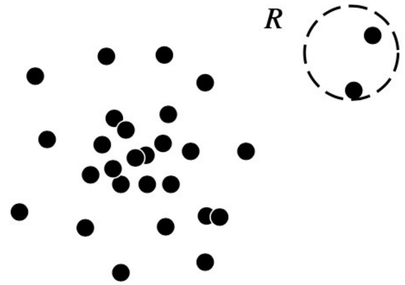
- Ex: Frauds, network intrusions, or system failures.
- Often used interchangeably with ***outliers*** but some define them differently.
- Not the same as ***noises***

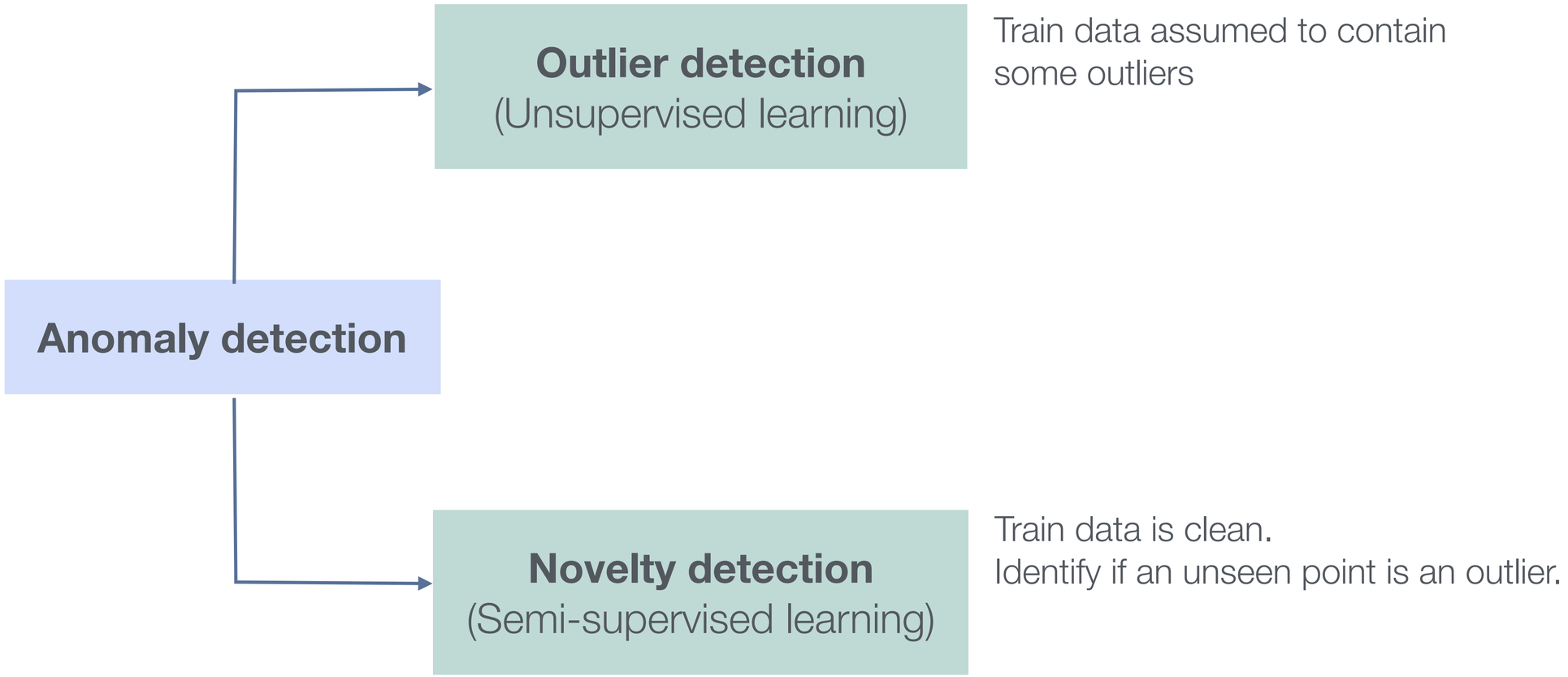
Anomalies (or outliers) should be detected and removed if developing supervised models.



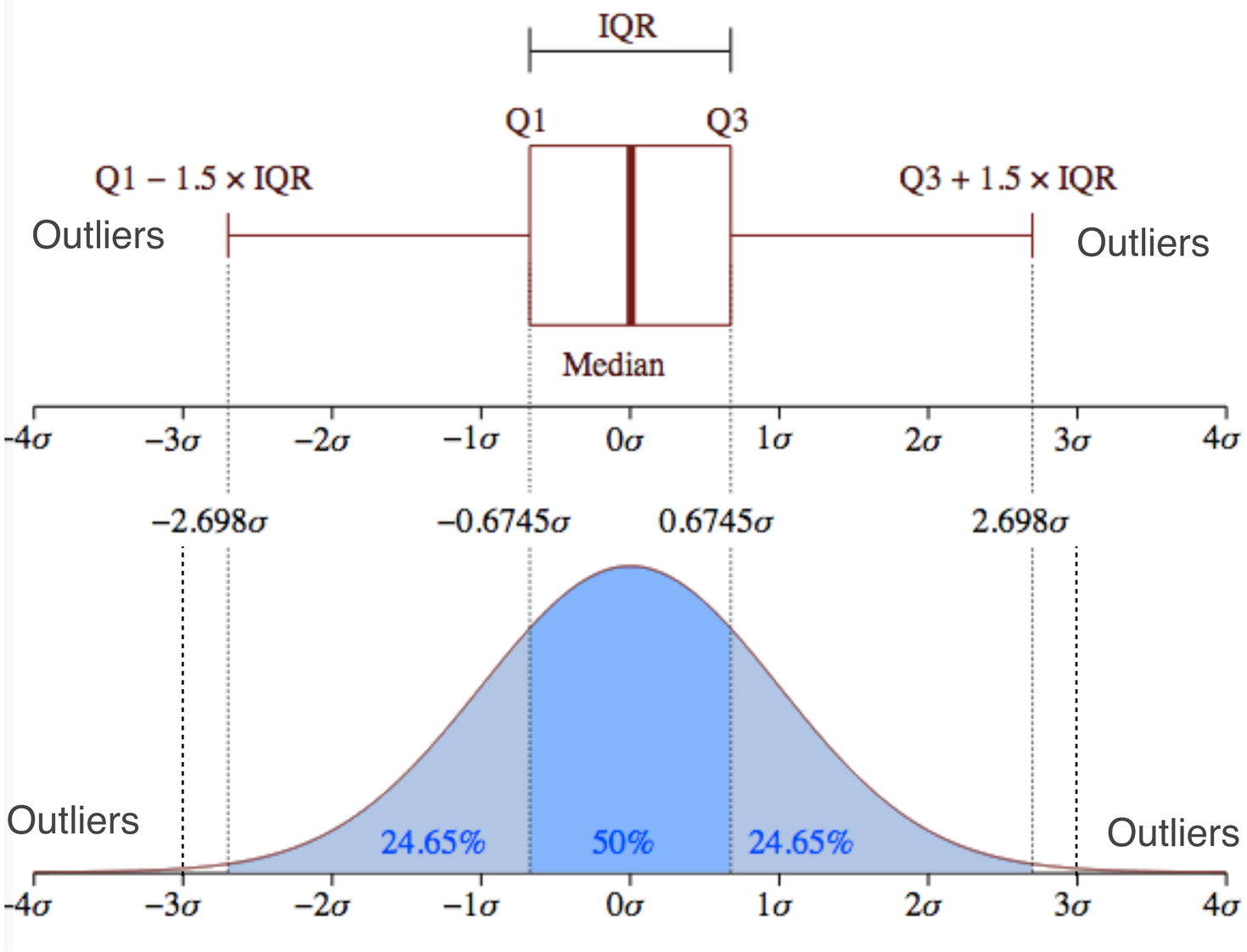







Anomaly detection
Outlier detection
(Unsupervised learning)
Train data assumed to contain some outliers
Novelty detection
(Semi-supervised learning)
Train data is clean.
Identify if an unseen point is an outlier.

Recap: Box Plot, Z-Score, and Outliers

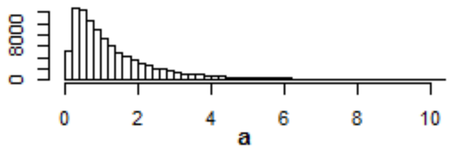


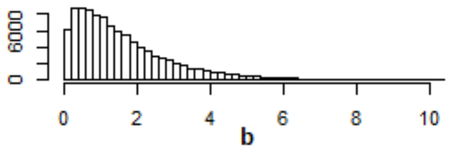
$$z_i = \frac{x_i - \bar{x}}{s}$$

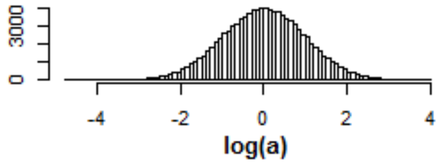
Unusual observations	Outliers
$3 > \|z\| > 2$	$\|z\| > 3$

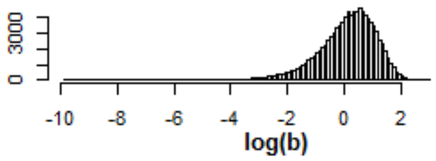
Fixing Skewed distribution

Data should not be too skewed when using box plot or z-scores to identify outliers.

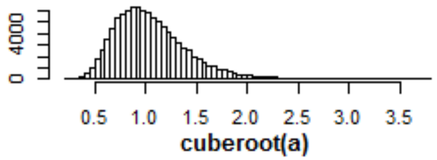


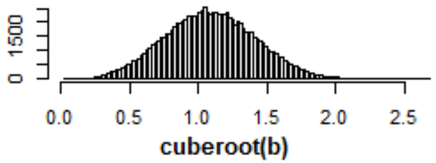






Log transformation, e.g., $\log_e, \log_2, \log_{10}$





Power transformation, e.g, $\sqrt{x},\ x^{1/3},\ x^2$

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{if} \quad \lambda \neq 0 \\ \log(y) & \text{if} \quad \lambda = 0 \end{cases}$$

Box-cox transformation

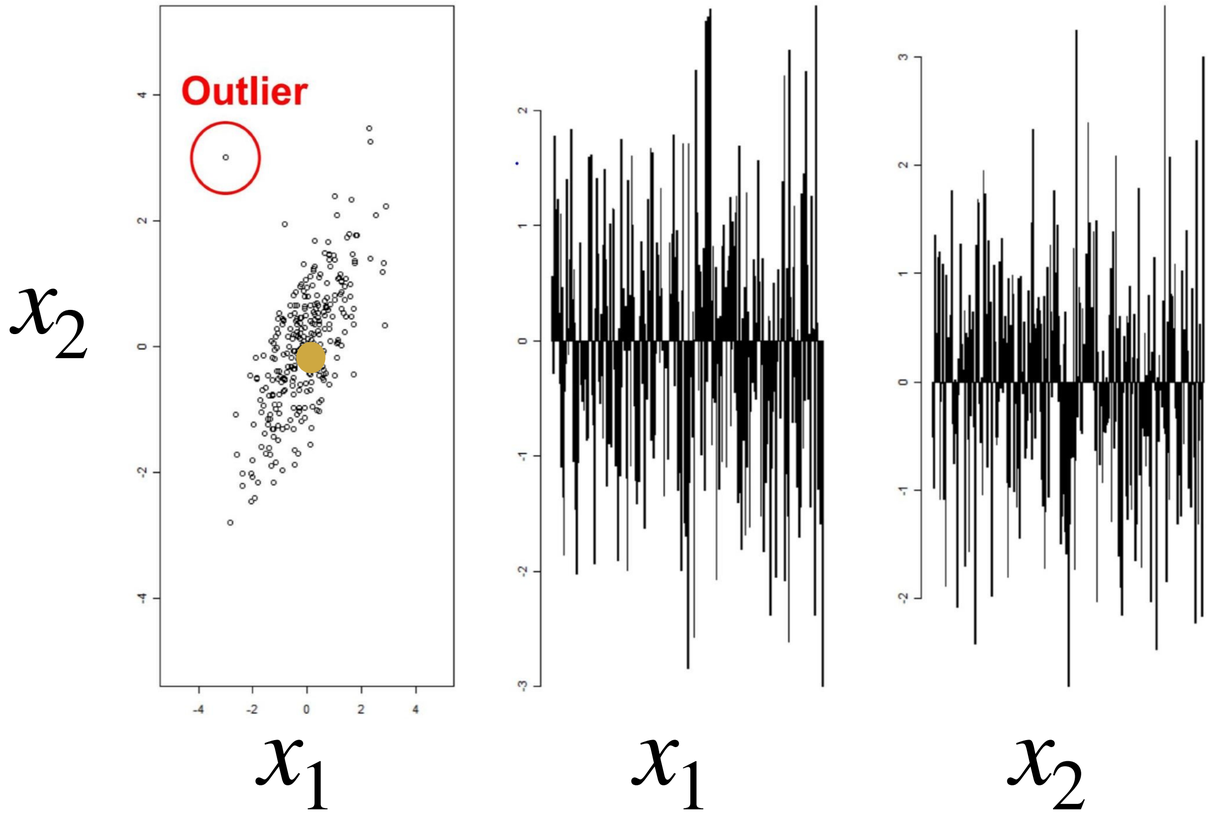
$(-5 \leq \lambda \leq 5)$

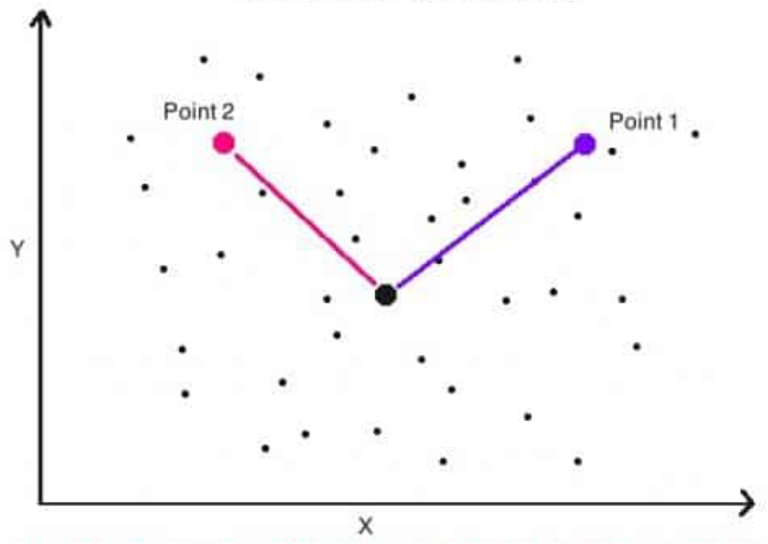
```
from scipy.stats import boxcox

df['boxcox_x'], _ = boxcox(df.x)
```

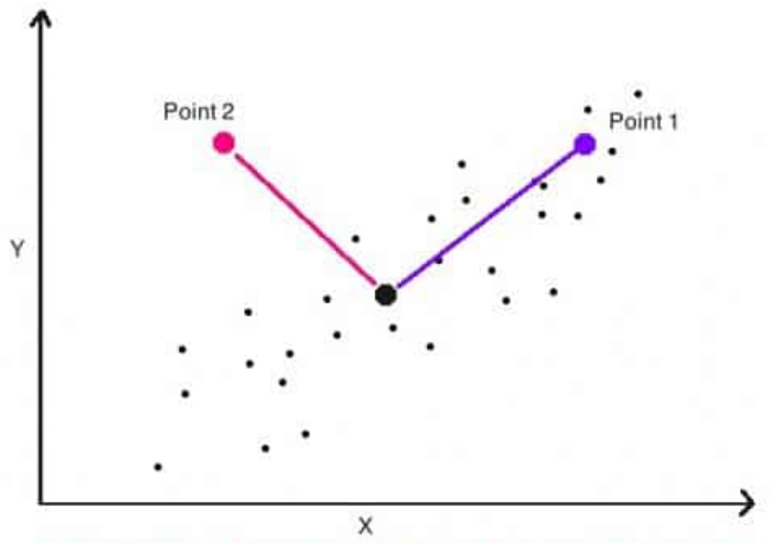
Multivariate Outlier

Applying univariate techniques to individual features in multivariate data does not work.

Euclidean distance from center could be used to identify outliers.




Point 2
Point 1
Y
X

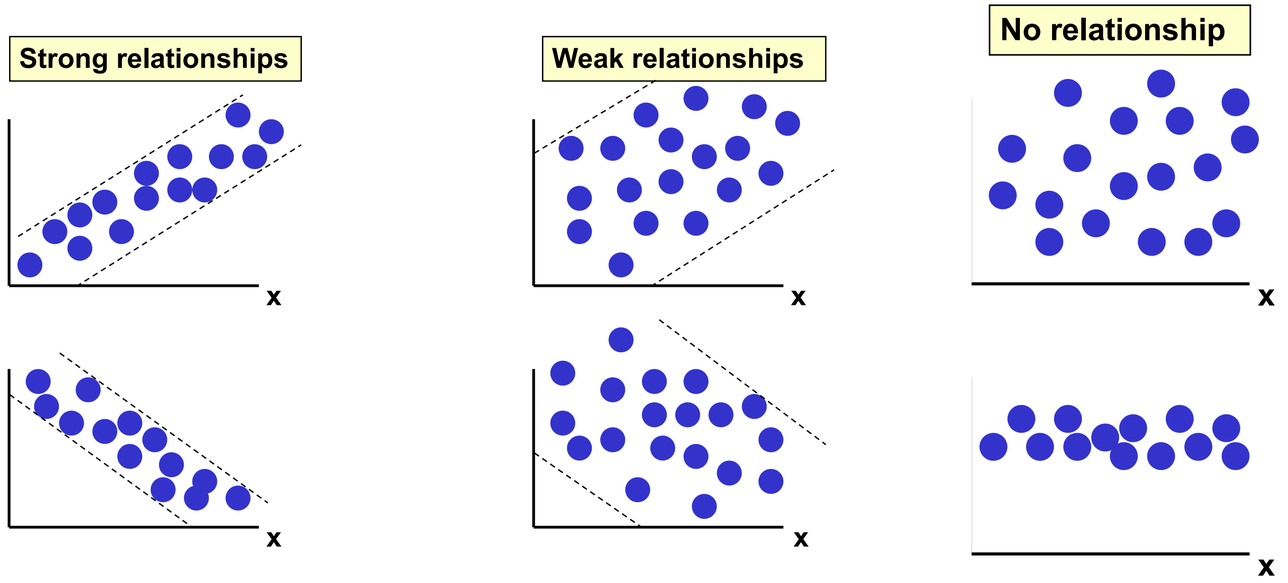

Point 2
Point 1
Y
X

Covariance

$$\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2]$$

Features

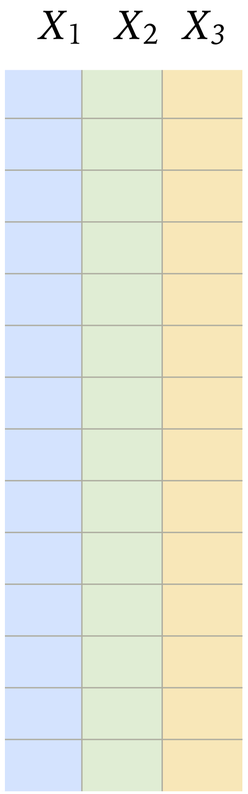
X_1 X_2



$$\mathbb{C}\mathrm{ov}(X_1, X_2) \triangleq \sigma_{12} = \frac{\sum_i (x_{1i} - \overline{x}_1)(x_{2i} - \overline{x}_2)}{n}$$

Covariance Matrix

Represent all pair-wise linear correlation relationships in dataset.



$$\mathbf{\Sigma} = \begin{bmatrix} \mathbb{C}\text{ov}(X_1, X_1) & \mathbb{C}\text{ov}(X_1, X_2) & \mathbb{C}\text{ov}(X_1, X_3) \\ \mathbb{C}\text{ov}(X_2, X_1) & \mathbb{C}\text{ov}(X_2, X_2) & \mathbb{C}\text{ov}(X_2, X_3) \\ \mathbb{C}\text{ov}(X_3, X_1) & \mathbb{C}\text{ov}(X_3, X_2) & \mathbb{C}\text{ov}(X_3, X_3) \end{bmatrix}$$

For two variables,

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix}$$

For three variables,

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{bmatrix}$$

Mahalanobis Distance (MHD)

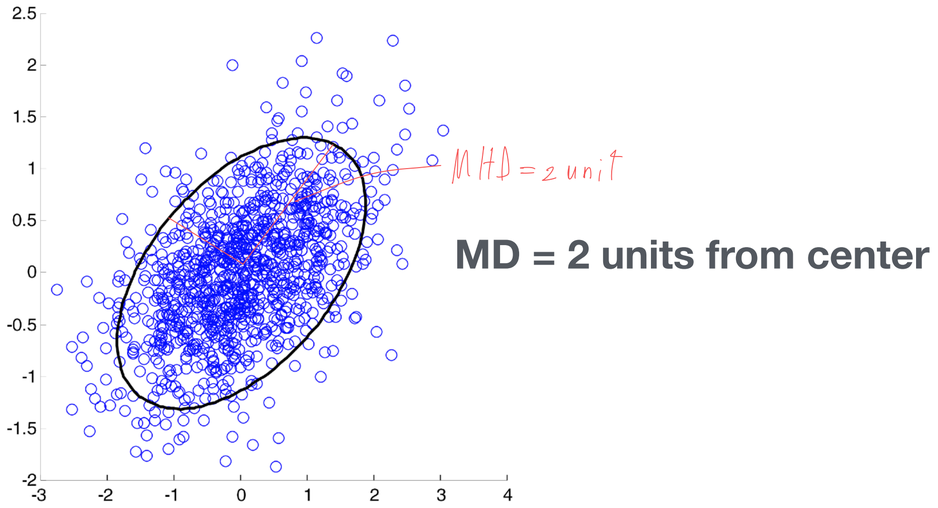
Assume multivariate normal data

Measure how many σ's away $\boldsymbol{x}_i$ is from $\boldsymbol{\mu}$ (centroid) considering the feature correlations:

threshold gaussian feature

$$\text{MHD}(\boldsymbol{x}) = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \mathbb{C}\text{ov}(X_1, X_1) & \mathbb{C}\text{ov}(X_1, X_2) \\ \mathbb{C}\text{ov}(X_2, X_1) & \mathbb{C}\text{ov}(X_2, X_2) \end{bmatrix}$$

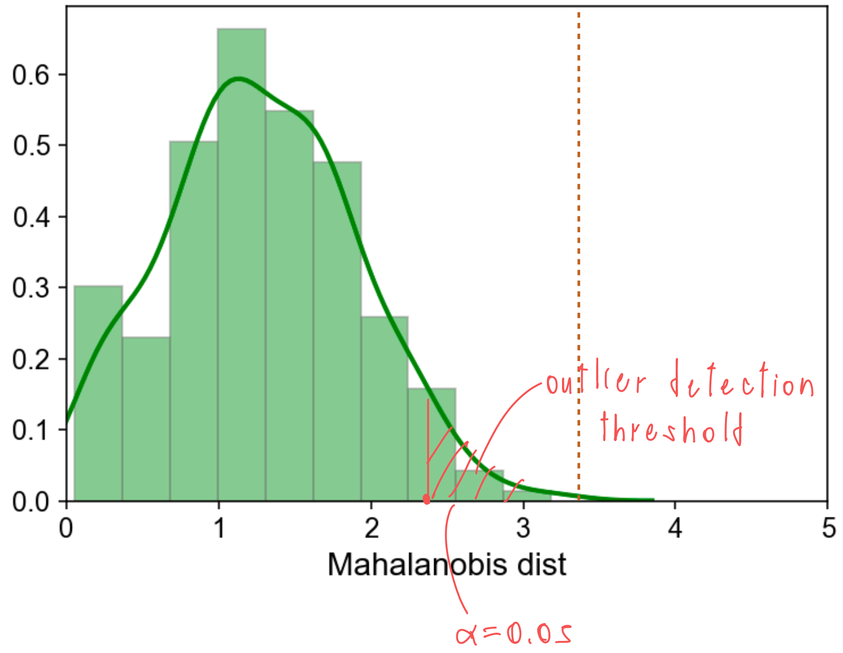


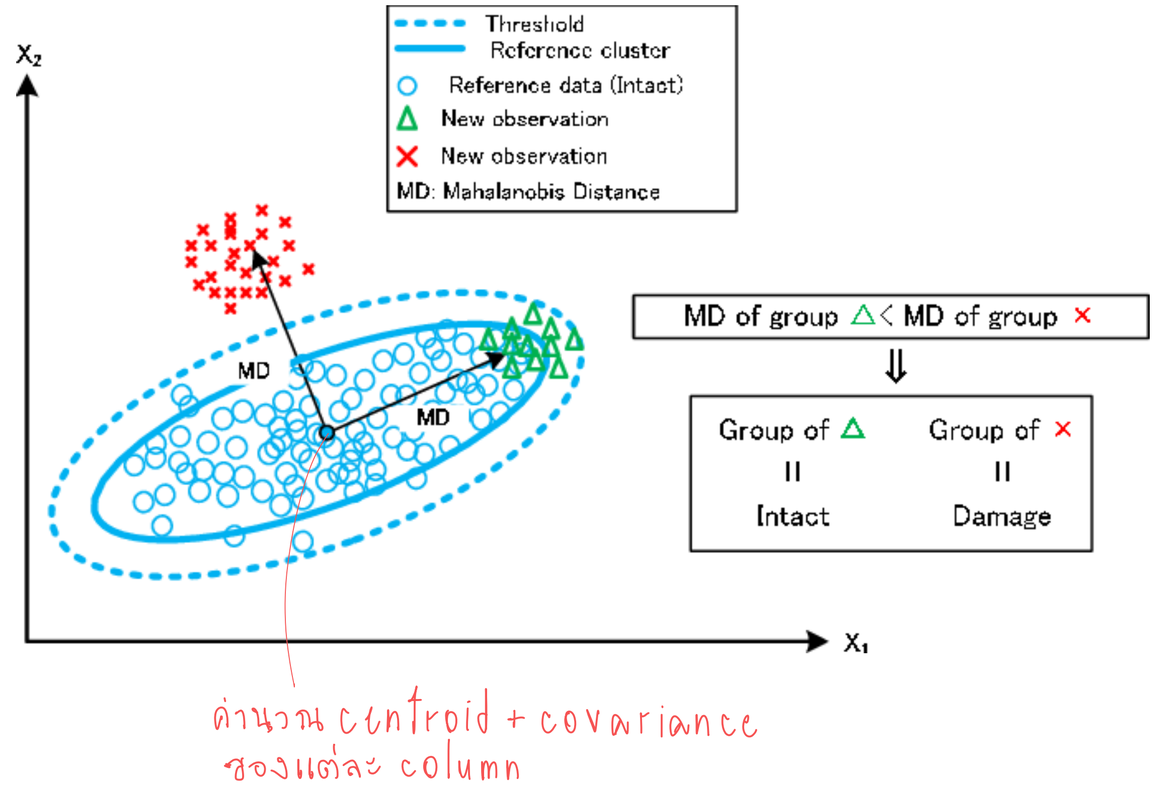
For uncorrelated data, MHD equals the euclidean distance of standarized features.

MHD Properties

data ต้องไม่ skewed

For multivariate normal data of k variables, the squared MHD $\sim \chi^2(k)$.



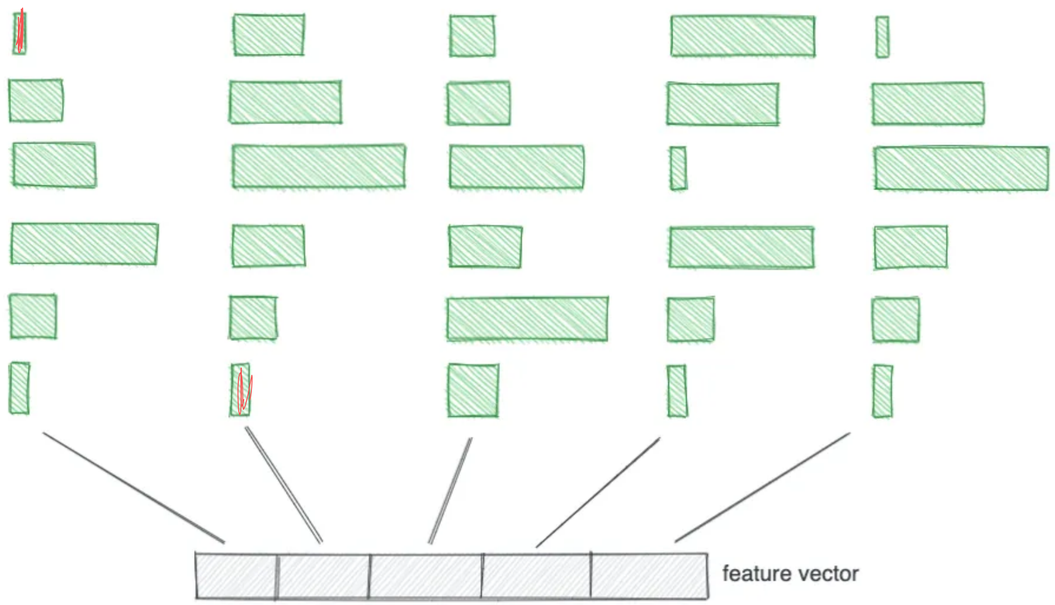


Histogram-based Outlier Scores (HBOS)

Create a normalized histogram (relative frequency counts) of each feature.

Calculate a score based on how likely a particular data point is to fall within the histogram bins.

An anomalous feature vector will occupy unlikely bins in one or several of its dimensions.



For a given data point of d features $\boldsymbol{x} = [x_1, x_2, x_3, \ldots, x_d]$, multiply the inverse of bin heights of each feature to obtain the HBOS as

$$\text{HBOS}(\boldsymbol{x}) = \log \prod_{i=1}^{d} \frac{1}{\text{hist}(x_i)} = \sum_{i=1}^{d} \log \frac{1}{\text{hist}(x_i)} + \alpha$$

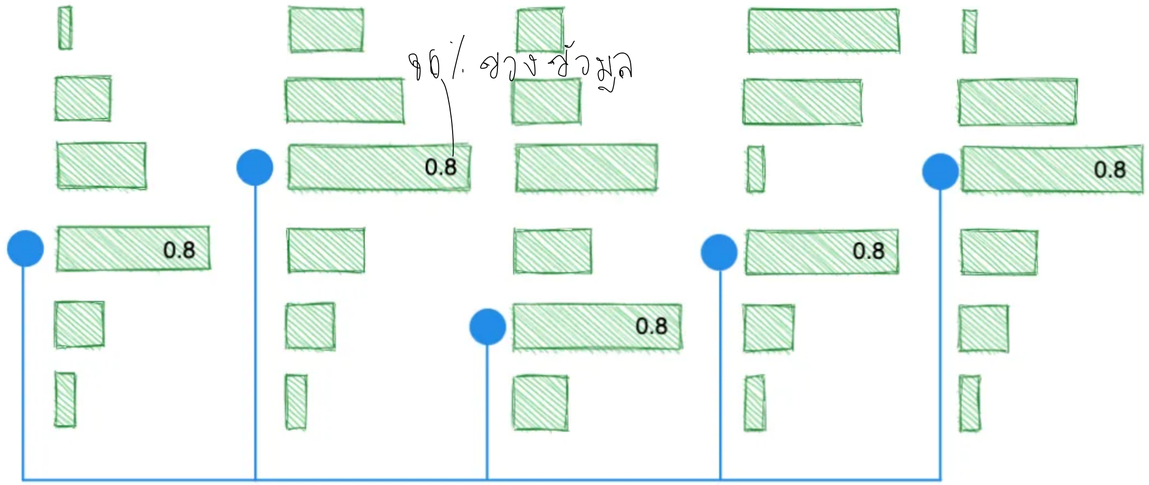
ป้องกัน overflow

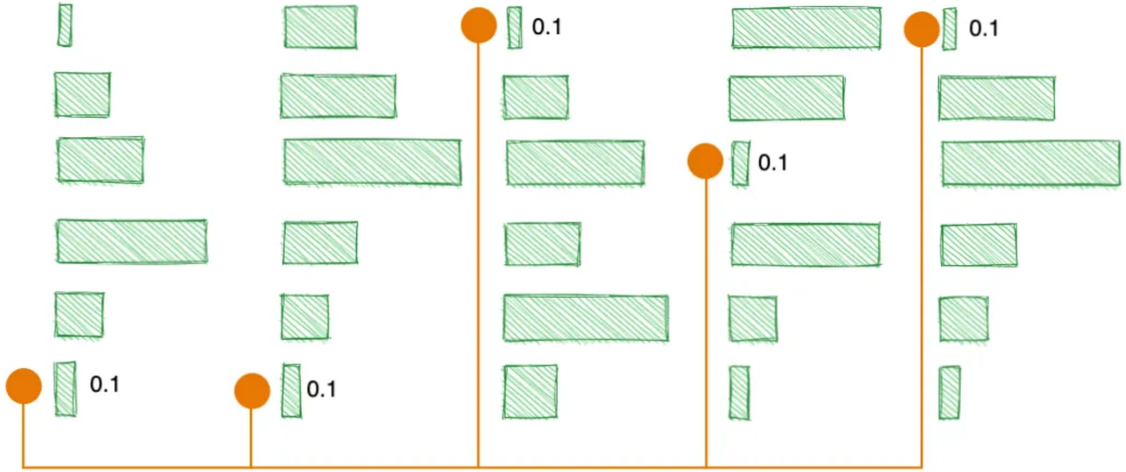
ระดับ ข้อมูล

Postive constant α can be added to $\text{hist}(x_i)$ to prevent overflow.

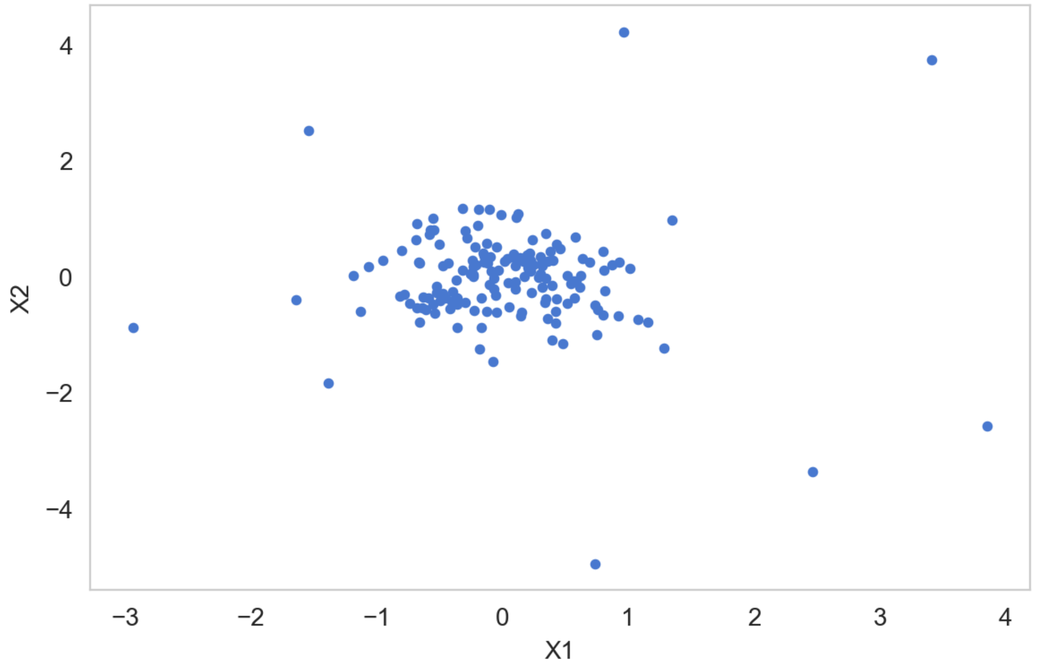
$$\text{HBOS}(\boldsymbol{x}_{normal}) = \log \prod_{i=1}^{5} \frac{1}{0.8} = 0.484$$

$$\text{HBOS}(\boldsymbol{x}_{anom}) = \log \prod_{i=1}^{5} \frac{1}{0.1} = 5$$





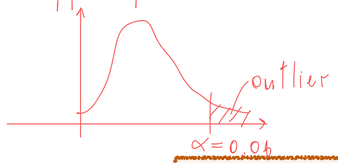
```
# Run this commmand if you have not done so.
!python -m pip install -U pip
!python -m pip install -U pyod

# Generate synthetic data
normal_data = np.random.normal(0, 0.5, (150, 2))
anomalous_data = np.random.uniform(-5, 5, (10, 2))

input_df = pd.DataFrame(np.vstack([normal_data, anomalous_data]), columns=['X1','X2'])
input_df.plot.scatter(x='X1', y='X2', figsize=(6,4), s=10);
plt.tight_layout();
```



```python
# Model fitting
from pyod.models.hbos import HBOS

n_bins = 20
hbos = HBOS(n_bins=n_bins, contamination=0.06)
hbos.fit(input_df)

# Identify outliers
y_scores = hbos.decision_function(input_df)
y_pred = hbos.predict(input_df)
print(f'Outlier threshold: {hbos.threshold_:.3f}')

pd.DataFrame(np.stack((y_scores, y_pred), axis=1), columns=['Score', 'Prediction'])
```

expected outlier proportion in training data

6% upper percentile in χ^2

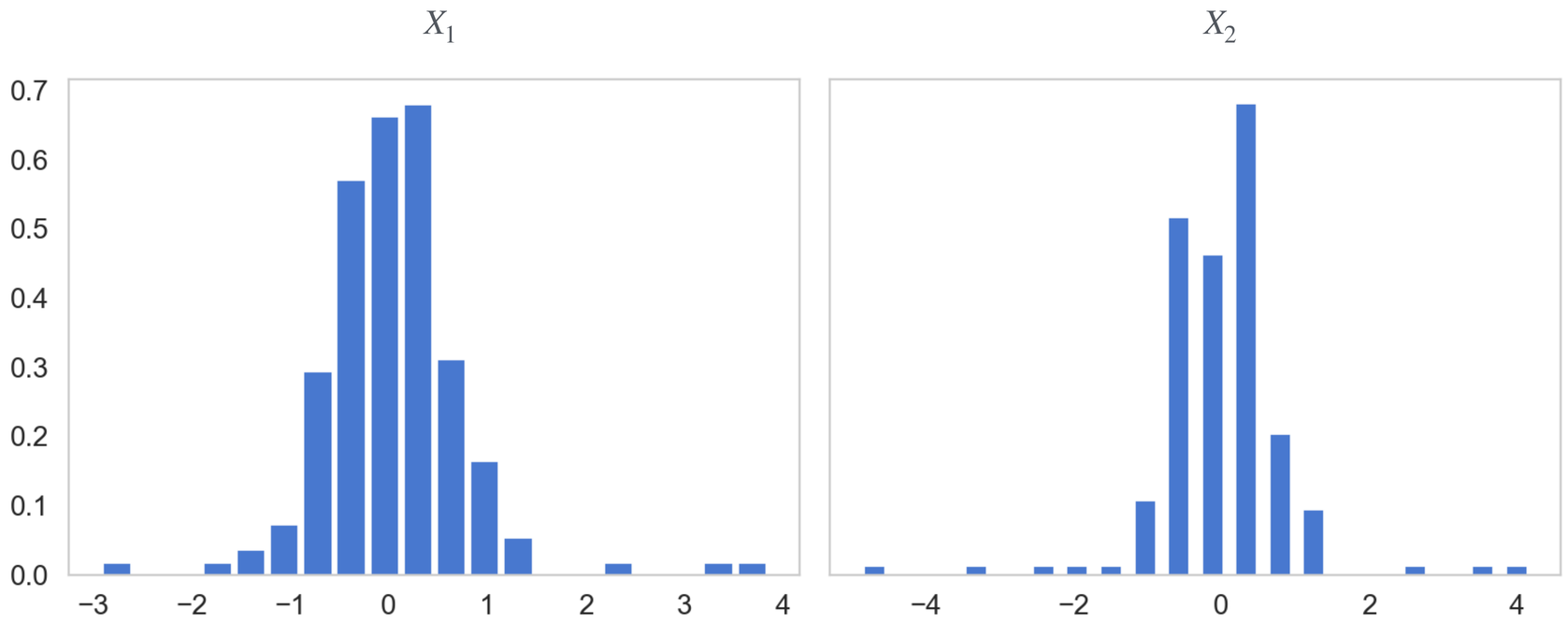



```
Outlier threshold: 4.062
```

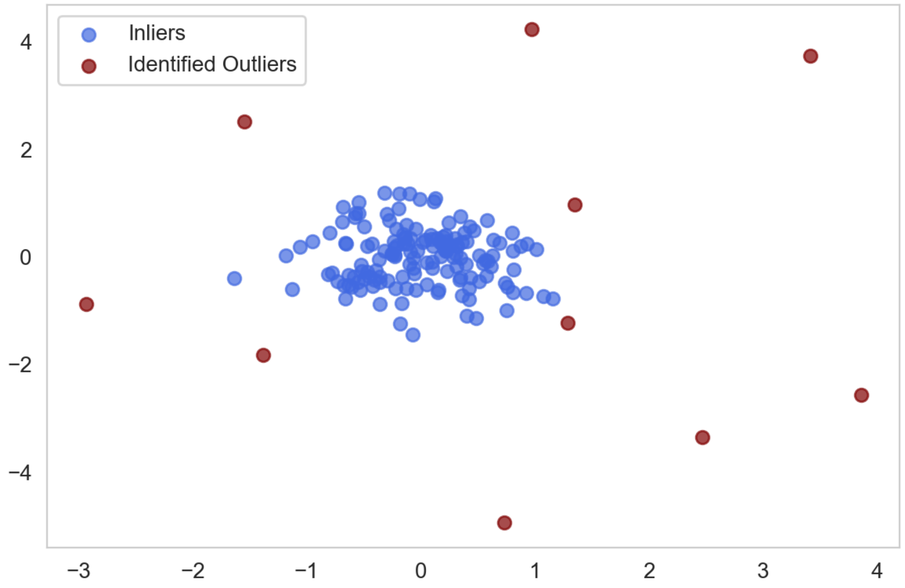
HBOS

	Score	Prediction
0	1.049848	0.0
1	1.969470	0.0
2	2.833767	0.0
3	3.533437	0.0
4	6.007077	1.0
...	...	...
155	5.049545	1.0
156	6.007077	1.0
157	3.528013	0.0
158	3.771829	0.0
159	6.215586	1.0

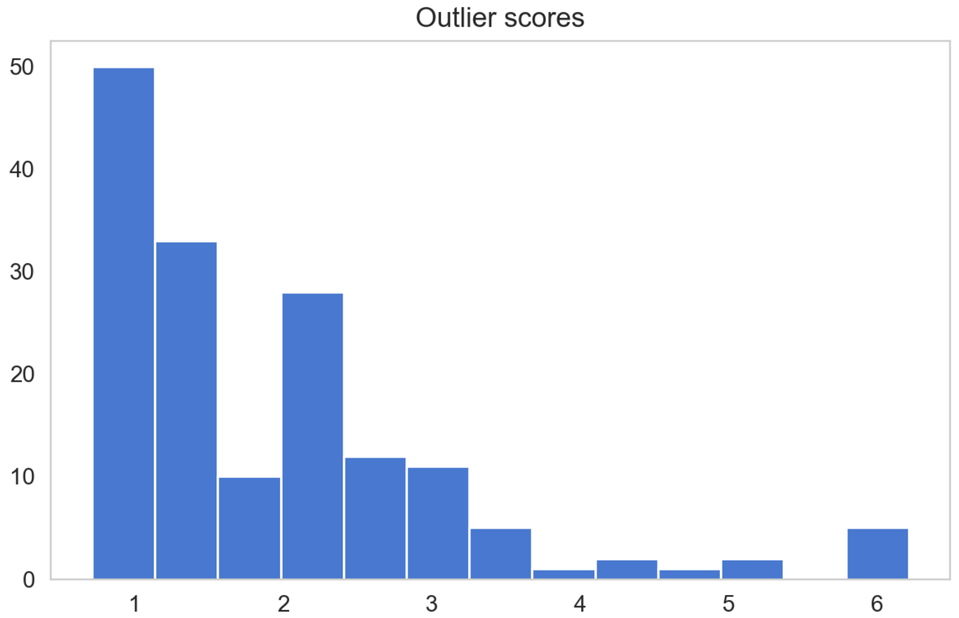
threshold ≈ 6



```python
# Extract inliers and outliers to visualize
outliers = input_df[y_pred==1]
inliers = input_df[y_pred==0]

plt.figure(figsize=(6, 4));
plt.scatter(inliers.X1, inliers.X2, label='Inliers', color='royalblue', alpha=0.7);
plt.scatter(outliers.X1, outliers.X2, label='Identified Outliers', color='maroon', alpha=0.7);

plt.legend();
plt.tight_layout();
```



Proper decision threshold can be determined from the histogram of outlier scores.

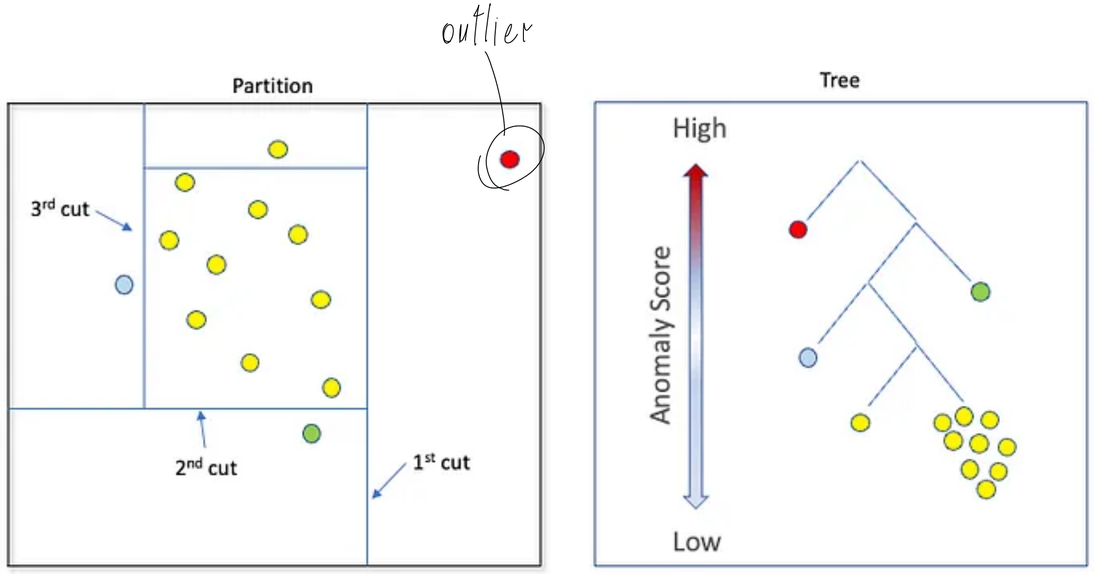


Isolation Forest

Recursively split the data points into halves such that each observation gets isolated from the others to obtain an ***isolation tree***.

- No target variable to guide the splittings.

Because anomalies, if any, lie away from the cluster of data points, they would be easily isolated compared to regular data points.



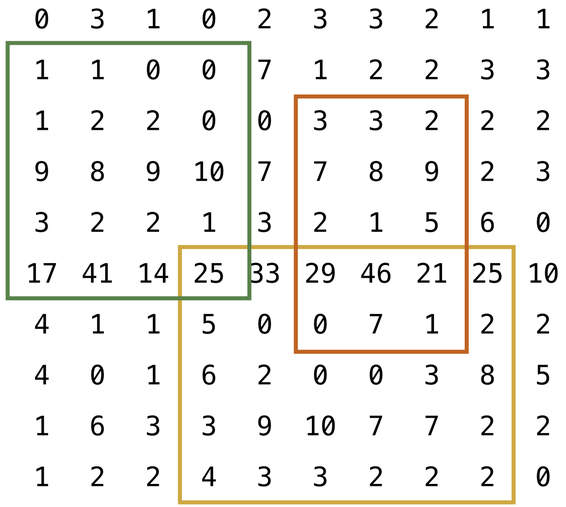
split ทีละครั้งจนกว่าจะมีตัวหลุดจากมา (random)

Generating Isolation Tree (iTree)

For a random subset of data of size ψ, recursively select a random feature and split the data at a random threshold of the feature.

Stop when

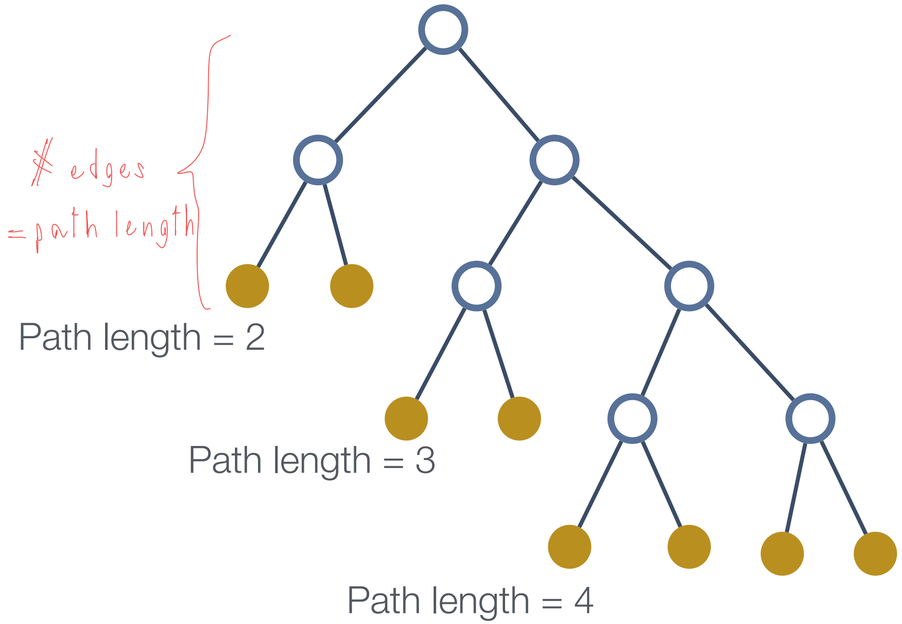
- the node has fewer than a predefined number of points (1 or 2), or
- the max depth (if defined) is reached.
- all data points at the node have the same value for every feature (rarely happens).



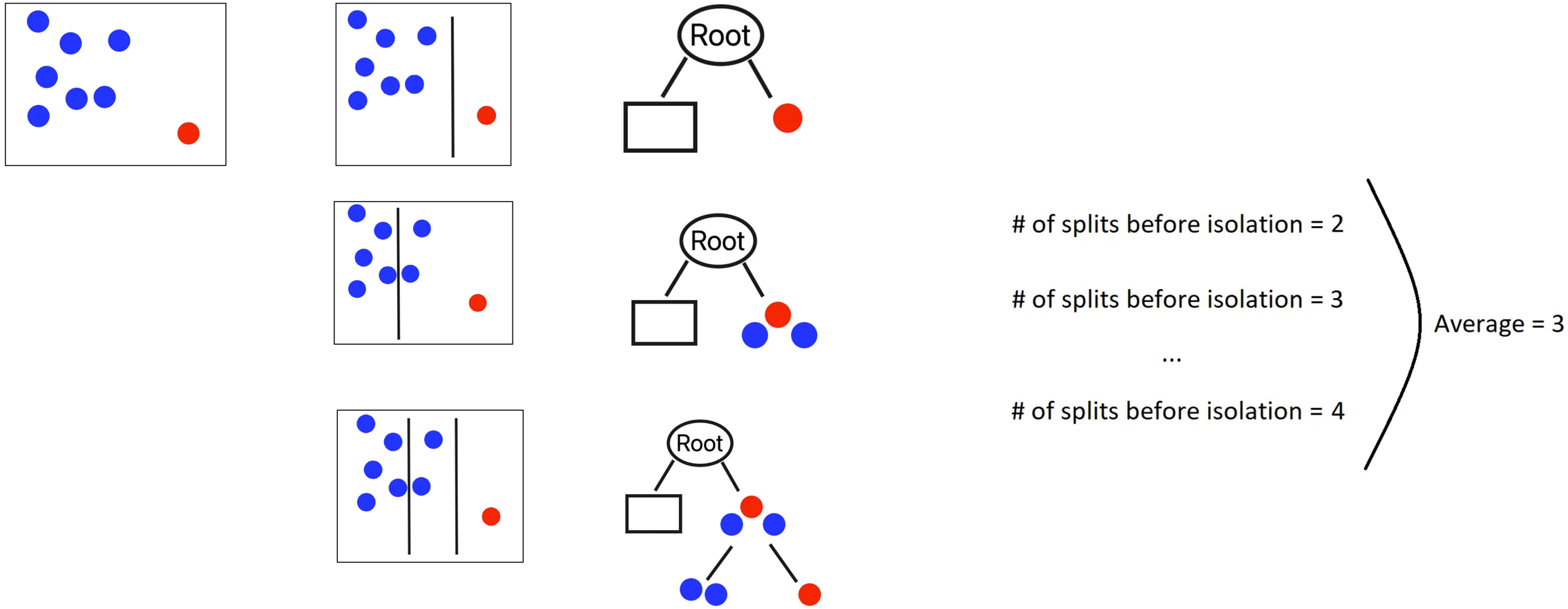
Path length (of the leaf node)

- # splits before isolation
- # edges from the root to the leaf node.

Random partitioning tend to produce noticeably shorter paths for anomalies.



Generate lots of iTrees from random subsamples of the dataset, and determine the average path length of x from those trees.



Anomaly Score

The anomaly score $s(\boldsymbol{x})$ of an instance $\boldsymbol{x}$ is computed based on the normalized average path length as

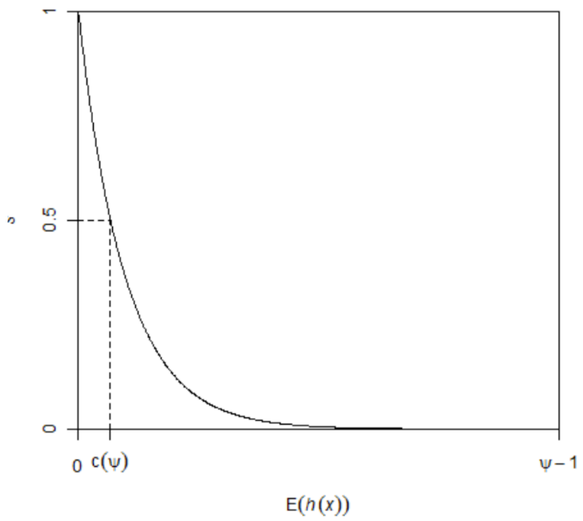
$$s(\boldsymbol{x}, \psi) = 2^{-\mathbb{E}\{h(\boldsymbol{x})\}/c(\psi)}$$

Expected value

Average path length for unsuccessful search in BST

$$c(\psi) = \begin{cases} 2H(\psi - 1) - 2(\psi - 1)/n & \text{for } \psi > 2, \\ 1 & \text{for } \psi = 2, \\ 0 & \text{otherwise.} \end{cases}$$

$$H(i) = \log(i) + 0.5772156649 \quad \text{(Harmonic number)}$$



- when $E\{h(\boldsymbol{x})\} \to 0$, $s \to 1$
- when $E\{h(\boldsymbol{x})\} \to \psi - 1$, $s \to 0$
- when $E\{h(\boldsymbol{x})\} \to c(\psi)$, $s \to 0.5$

Flaging anomalies:

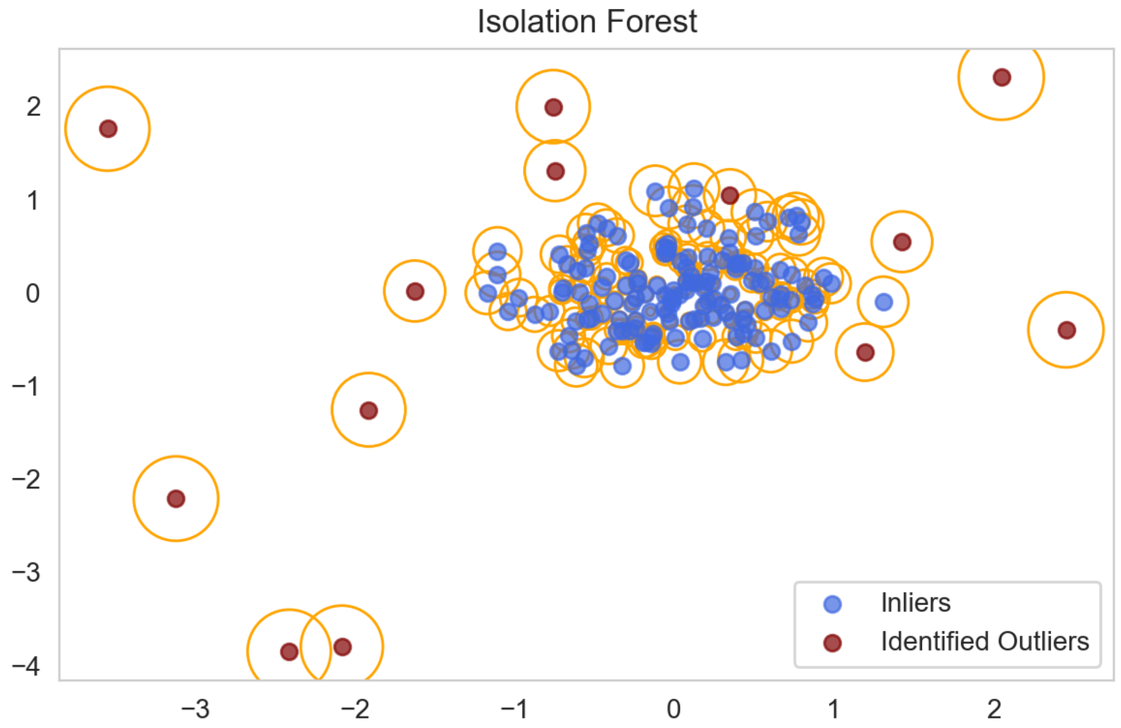
- Samples with $s \ll 0.5$ can safely be regarded as inliers.
- Those with $s \to 1$ are considered anomalies. So, the threshold 0.5 is used the default value to flag anomalies.

	X1	X2
0	0.658597	0.258447
1	-0.302188	-0.031452
2	-0.423834	-0.228524
3	0.107967	0.210495
4	-0.080863	-0.188804
...	...	...
155	2.348850	-1.713857
156	1.997780	3.835402
157	3.679225	2.552838
158	-2.769838	-2.276746
159	2.472803	-3.586134

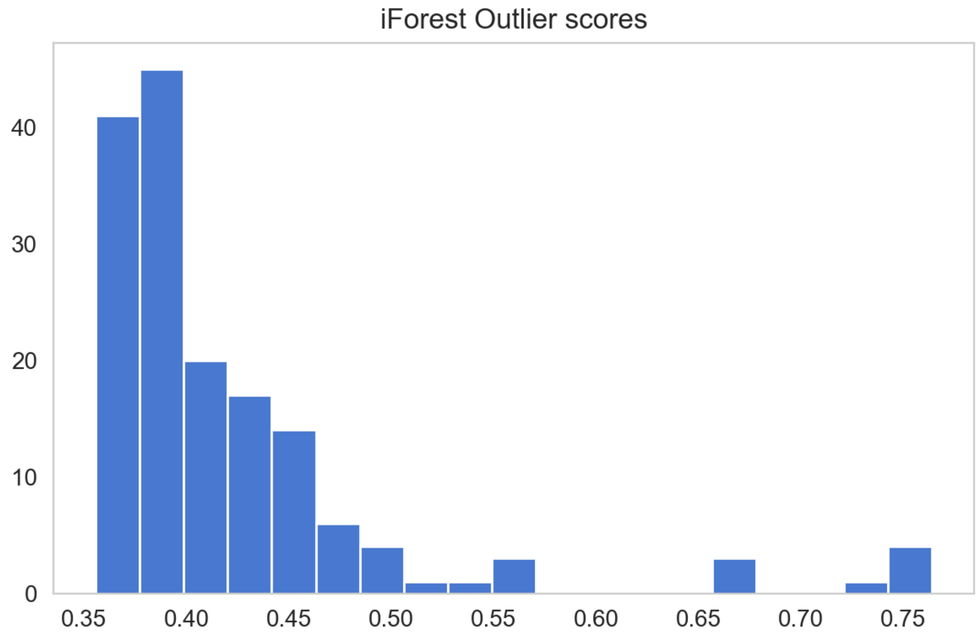
```python
from sklearn.ensemble import IsolationForest

isof_clf = IsolationForest(random_state=42, max_samples='auto')

y_pred = isof_clf.fit_predict(input_df)
y_pred

isof_scores = -isof_clf.score_samples(input_df)
isof_scores
```

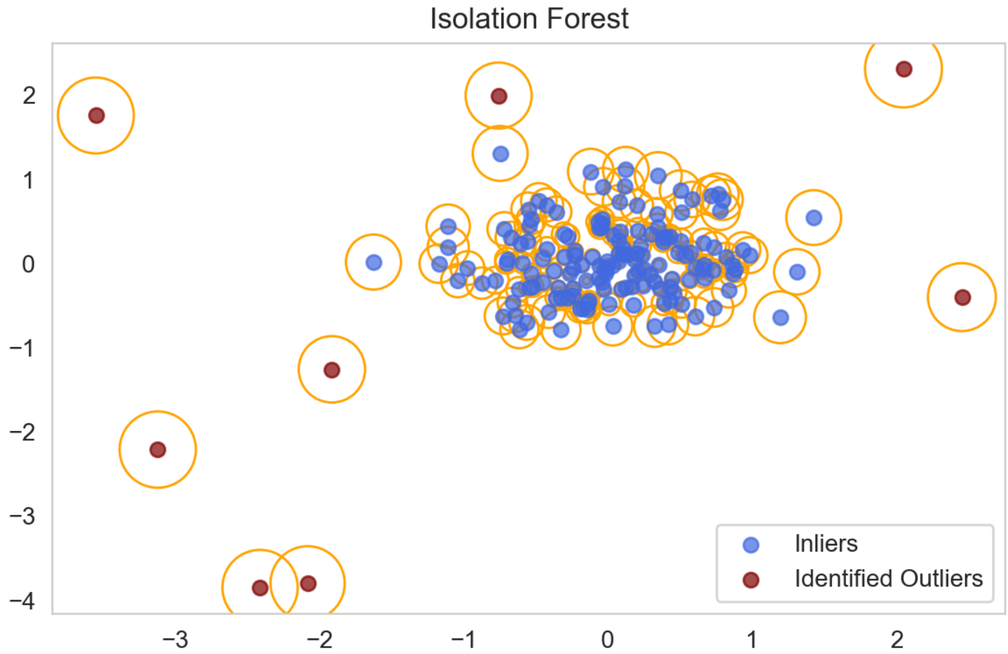
```
array([ 1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1,  1, -1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, -1,  1,
        1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, -1,  1,  1,  1,  1,  1,
        1,  1,  1, -1,  1,  1,  1, -1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1,  1,  1,  1,  1,  1, -1,  1,  1,  1,  1,  1,  1,  1,  1, -1,
       -1,  1,  1,  1,  1,  1,  1, -1,  1,  1,  1,  1,  1,  1, -1, -1, -1,
       -1, -1, -1, -1, -1, -1, -1])
array([0.37025239, 0.35500414, 0.37257833, 0.34428563, 0.35395134,
       0.37501819, 0.34604653, 0.41466438, 0.34570856, 0.40056365,
       0.35472459, 0.41605291, 0.44076102, 0.38355945, 0.36026852,
       0.38858909, 0.35161055, 0.40661835, 0.39762709, 0.36041055,
       0.51410441, 0.42485252, 0.36611215, 0.34229949, 0.34125295,
       0.41676231, 0.36275934, 0.40442795, 0.35457189, 0.39162121,
       0.406417  , 0.34350683, 0.57013196, 0.36768352, 0.4175088 ,
       0.4432247 , 0.37259086, 0.36747779, 0.37868943, 0.37579186,
       0.37496458, 0.36092084, 0.43225876, 0.36875275, 0.35315264,
       0.52669942, 0.48291653, 0.4179639 , 0.36182093, 0.37571487,
       0.36329652, 0.40156966, 0.4401895 , 0.38876945, 0.50168548,
       0.43763   , 0.35315952, 0.34594189, 0.50397794, 0.40475179,
       0.38859898, 0.34043254, 0.35935837, 0.43480998, 0.39691287,
       0.35387739, 0.47070482, 0.49572215, 0.41626064, 0.36419096,
       0.3529273 , 0.35285628, 0.40459626, 0.36342374, 0.34603931,
       0.47165731, 0.36960258, 0.40521993, 0.36068151, 0.36051035,
       0.35738294, 0.35150052, 0.39935876, 0.38393749, 0.3579096 ,
       0.34913429, 0.39030963, 0.45731172, 0.36851279, 0.42071579,
       0.3875913 , 0.34941702, 0.37074301, 0.35209068, 0.36697064,
       0.37357463, 0.34188044, 0.36585039, 0.35944466, 0.37505266,
       0.34944619, 0.350783  , 0.35688814, 0.34919693, 0.34634385,
       0.36182377, 0.34913429, 0.39954603, 0.37487875, 0.3484436 ,
       0.37561579, 0.38199552, 0.41628639, 0.36740863, 0.46588514,
       0.35272779, 0.35415733, 0.34441669, 0.35487333, 0.37277933,
       0.37129682, 0.35751645, 0.39858671, 0.37364978, 0.34142944,
       0.39171085, 0.36203273, 0.51065161, 0.36605967, 0.36345834,
       0.40051303, 0.35838834, 0.34706278, 0.37511983, 0.35109814,
       0.50021753, 0.53410632, 0.37417918, 0.37964732, 0.37962497,
       0.36840176, 0.37764761, 0.34979657, 0.51198087, 0.367814  ,
       0.36130796, 0.38061068, 0.37068391, 0.41747844, 0.33838079,
       0.68442624, 0.74675626, 0.72313369, 0.68289837, 0.65062051,
       0.68892392, 0.70407761, 0.71838236, 0.72591369, 0.76477136])
```

} > 0.5 ⇒ outlier



From the histogram of outlier scores, a more appropriate anomaly threshold can be determined.



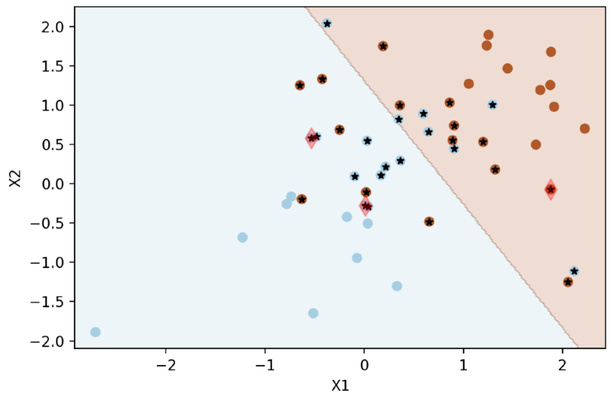


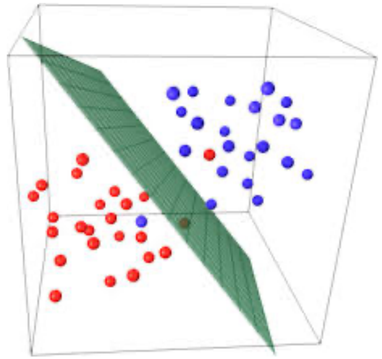
One-Class SVM

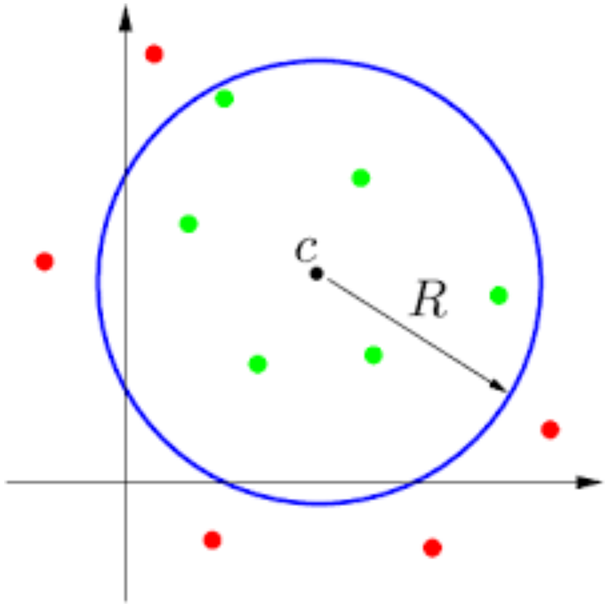
A regular SVM algorithm tries to find a hyperplane that best separates two classes of data points.

For one-class SVM,

- Training data is assumed to contain only normal data. (inlier)
- Create a hypersphere with smallest radius or boundaries that attempt to "enclose" the data.
  (ทรงกลมที่ครอบ inlier เอาไว้)





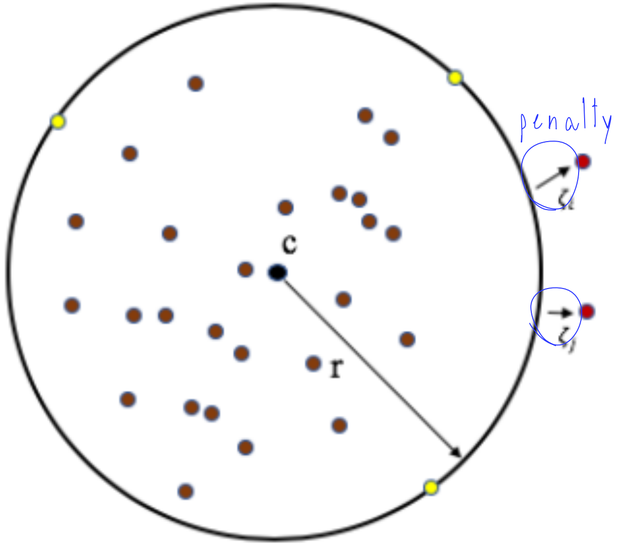


The optimal hypersphere with centre c and radius r with flexibility to tolerate outliers to an extent is obtained from

error budgets

$$\min_{r,c,\zeta} r^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\zeta_i$$

$$\text{subject to, } ||\Phi(x_i) - c||^2 \leq r^2 + \zeta_i \quad \forall i = 1, 2, \ldots, n$$



% outlier

The hyperparameter "nu" $0 < \nu < 1$ trades-off the model's ability to categorize data as normal or abnormal.

- Smaller ν = Larger sphere ⟶ less point encloses in sphere
- Should be set to % suspected outliers.

hypersphere boundaries

In Scikit-learn, outlier scores of OCSVM are the value of $\Phi(x_i)$, the signed distance to the separating hyperplane.

- Positive values -> inliers
- Negative values -> outliers

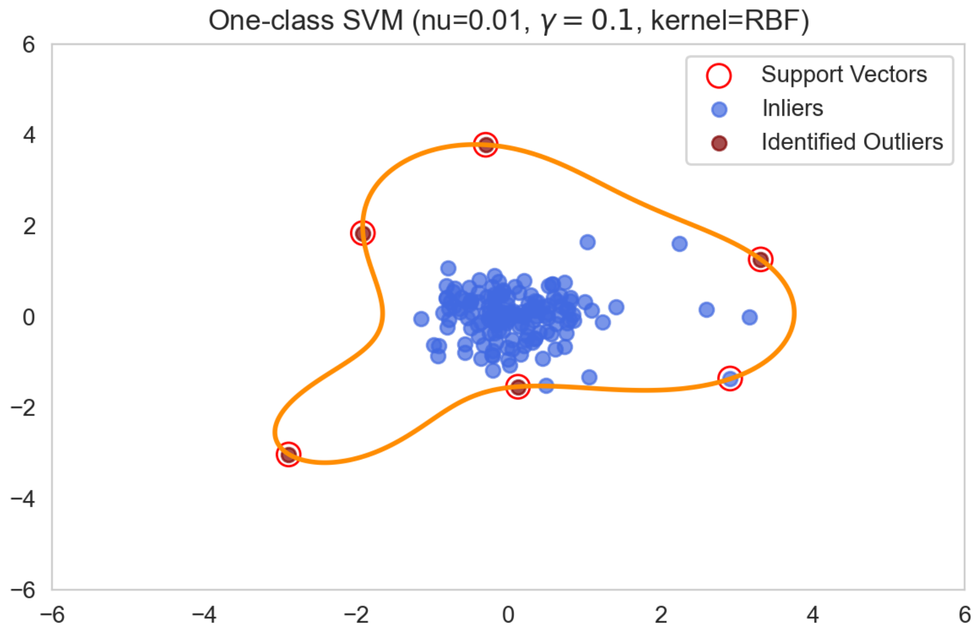
```
class sklearn.svm.OneClassSVM(*, kernel='rbf', degree=3, gamma='scale',
coef0=0.0, tol=0.001, nu=0.5, shrinking=True, cache_size=200, verbose=False,
max_iter=-1)
```

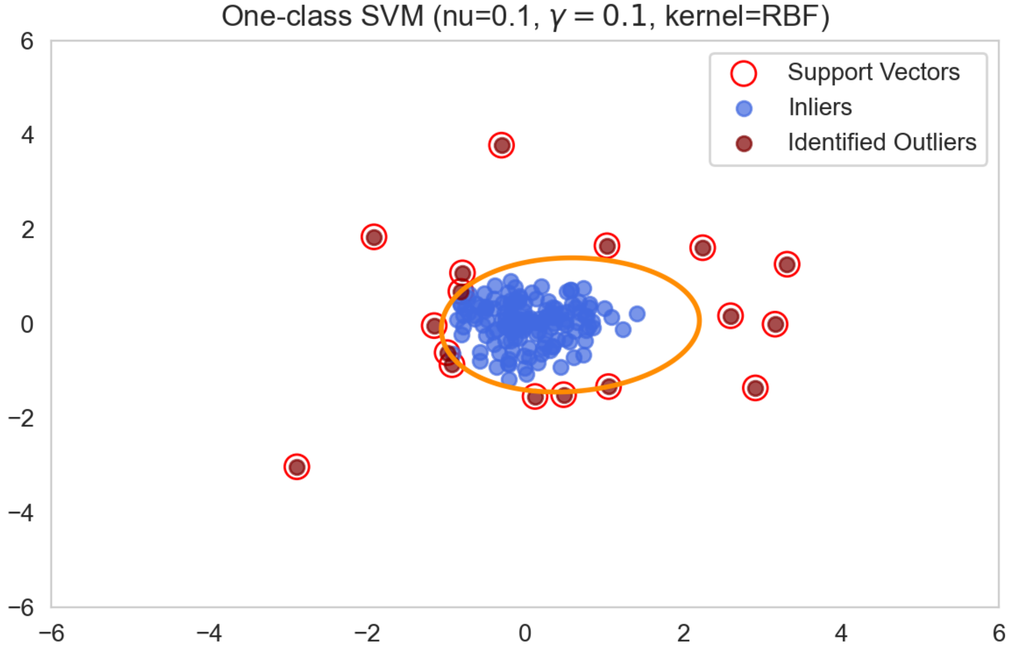
[source]

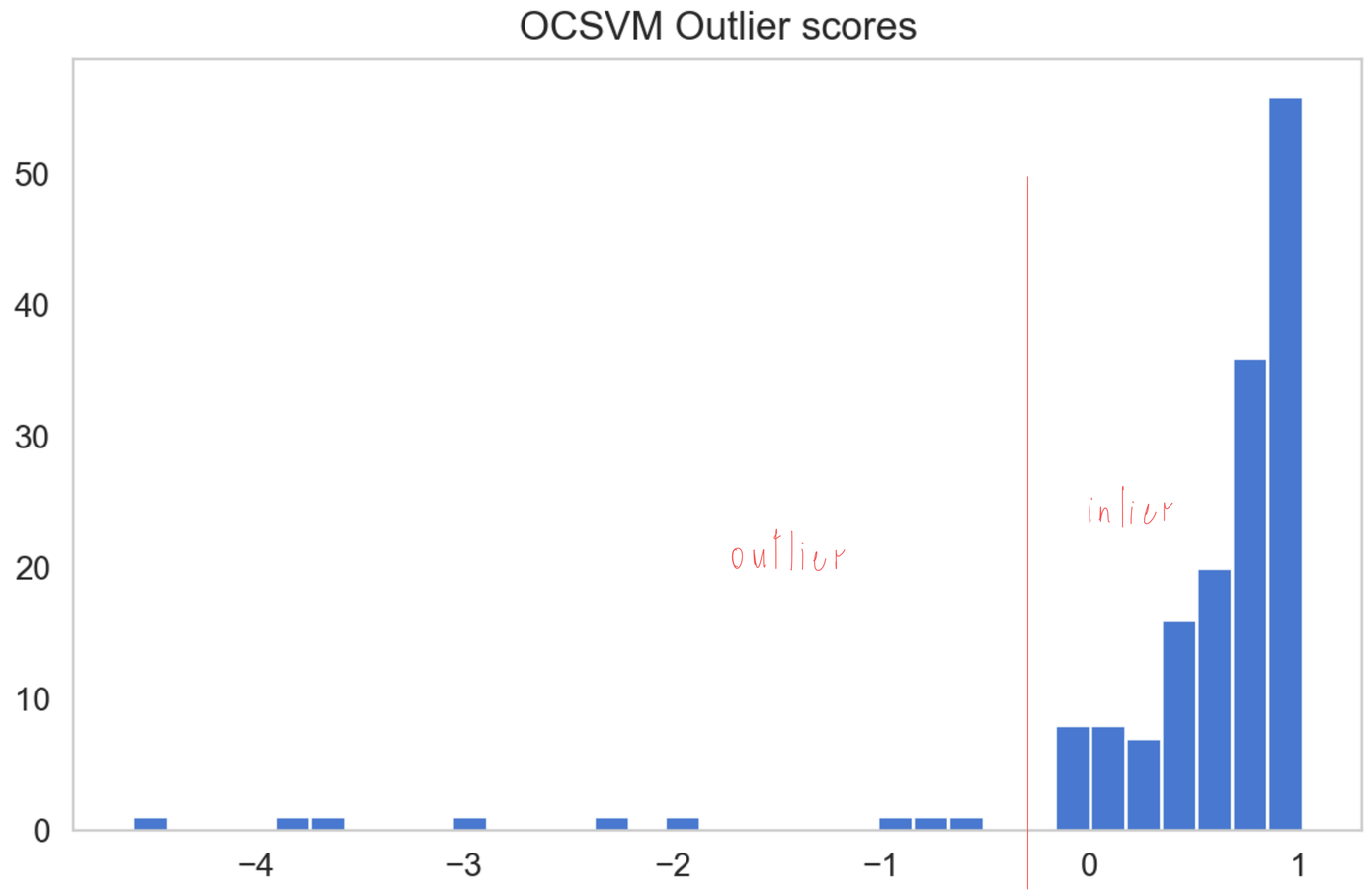
```python
from sklearn.svm import OneClassSVM

KERN = 'RBF'
NU, GAMMA = 0.01, 0.1

ocsvm_clf = OneClassSVM(nu=NU, kernel=KERN.lower(), gamma=GAMMA)
ocsvm_clf.fit(input_df)

y_pred = ocsvm_clf.predict(input_df)
ocsvm_scores = ocsvm_clf.decision_function(input_df)
```

	X1	X2
0	0.658597	0.258447
1	-0.302188	-0.031452
2	-0.423834	-0.228524
3	0.107967	0.210495
4	-0.080863	-0.188804
...	...	...
155	2.348850	-1.713857
156	1.997780	3.835402
157	3.679225	2.552838
158	-2.769838	-2.276746
159	2.472803	-3.586134

small nu






OCSVM Outlier scores
50
40
30
20
10
0
−4
−3
−2
−1
0
1
outlier
inlier

ส่วนเส้น

Latent-Variable-based Approaches

เหมาะกับข้อมูลที่มี dim เยอะๆ

n rows of observations/ data points/ Users

Normal data with high dimension d

Dimensionality reduction algorithm

e.g., PCA, Autoencoder

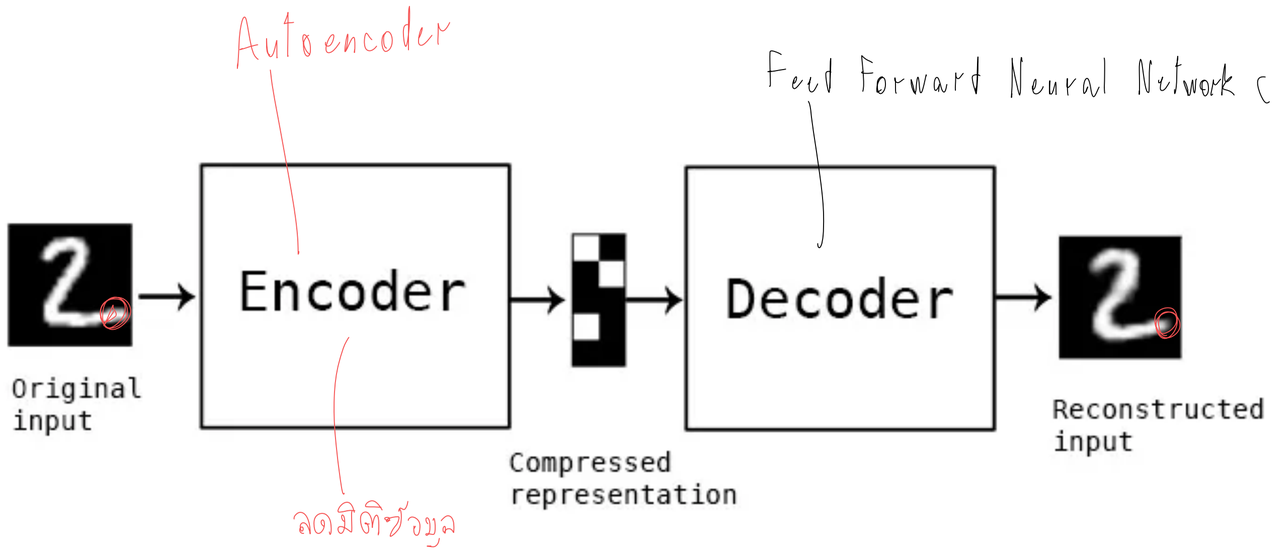
Latent-variable Model

ข้อมูลที่ถูกลดมิติ

Anomaly detection

Autoencoders (AE) FFNN

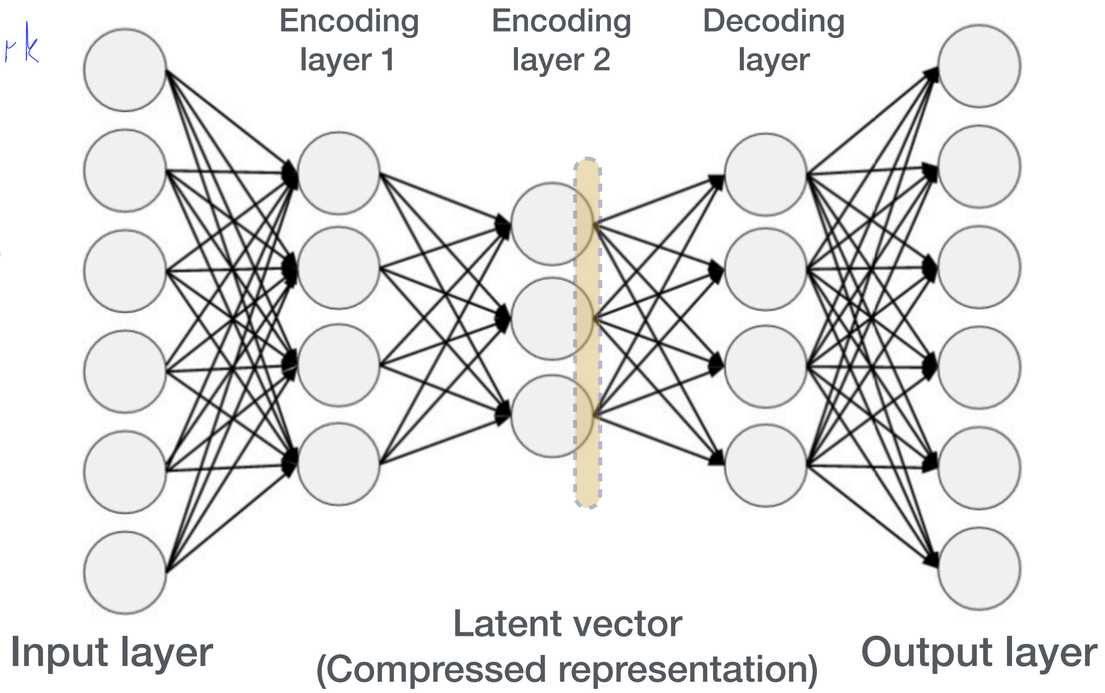
Feed-forward network intends to reproduce the input or its variants by using encoder-decoder architecture.

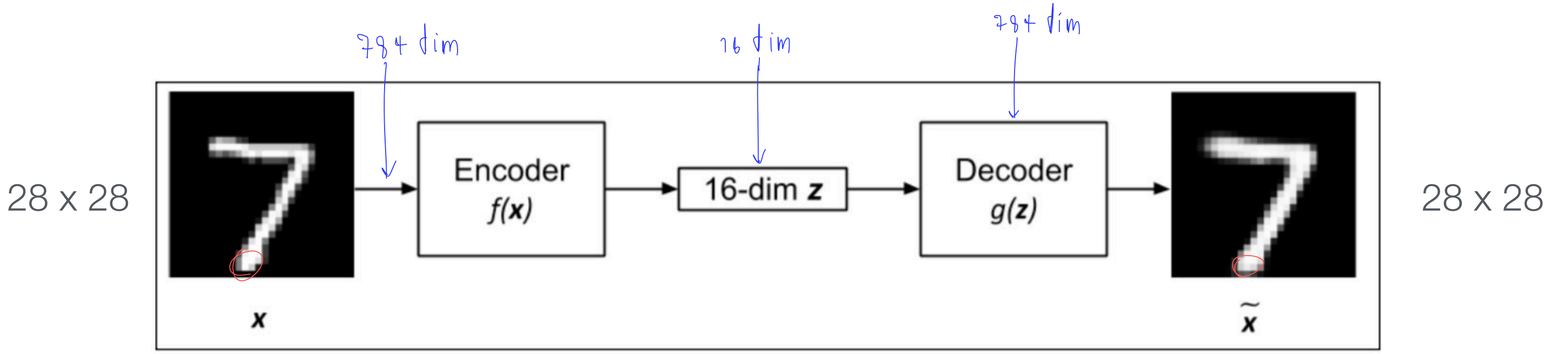


Autoencoder for Dimensionality Reduction

Feed-Forward Neural Network

The network can be shallow or deep, depending on # hidden layers in the encoder/decoder.





Anomaly Detection Lab

(Group of 4 to 6)