* Outliers Detection - Box plot

1. Median 25%
2. 75% population.
3. Inter Quartile
4. IQR

- outlier is an observation ie; unlike the other observation.

Caused by : measurement or i/p error
Data corruption
True outlier observation.

May cause problem during model fitting.

noise : random behaviour / value scattered randomly in dataset

outlier : data follows trend, but some points do not.

Possible Solutions:

A model that can handle outliers, max margin classifiers SVM.

Seaborn feature.

import seaborn as sns

sns.boxplot (data=df, x = column, ax= axs [i, 1], color = color

Step 1: Ascending

Step 2: Median
Q_1
Q_3
IQR

Step 3: Max, Min

Imputing Missing Values

↳ Mean Value (of a column)
↳ Median Value
↳ Mode
↳ Random Sample imputation

Naive Bayes model, Nearest neighbors.

* Limited Quantity of Training Data - Problems

- # of samples in training data (x_i) >>> # of parameters in training data (Q)

- Overfitting : Happens in case of large K & small n.

- Poor Generalization

  Cat dog identification
  ↓ sitting ↓ standing

  Not always!

  Test data: Cat standing identifies as dog.
  (Picks up the noise).

* Data Augmentation → slightly modify the image.
(Perturbations)

original learns more about distribution of data image.

Augment your dataset by creating additional training examples through rotation, flipping, cropping.

* Feature engineering

↳ which features are actually contributing to the ~~best~~ model.

- High quality features can helps your model learn from the available data more effectively.

* Reduction of parameters

* Alternate techniques

↳ regularization

↳ Semi-supervised learning.

(data cleaning

Data Transformation

Linear Transformation;

$X \; a = y$

matrix vector

classific^n → not easy to find decision boundary

Transforms into a space where it can easily be distinguished.

Linear Transformation:

function that maps i/p vector → o/p vector.

Rotation $X = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$

Scaling

Shear → change the orientation of vector.

(If we can't find decision boundary in one space, it then makes/transforms into another space where it can be easily distinguished - decision boundary).

Support Vector

Neural networks: Complex datasets

how to transform data in another basis that could simplify the data.

Same info, but shown in diff way

we can learn about decision boundary, data properties

→ A line should remain a line after transformation.

→ And origin will be fixed.

→ Dist b/w the grid lines should remain equidistant.

Properties: $T(\vec{u}+\vec{v}) = T(\vec{u}) + T(\vec{v})$

$T(c\vec{u}) = cT(\vec{u})$

$T(0) = 0$

① Data Normalization

- Some features dominating distances

- Models don't realize that datapoints may or may not be of same units.

Every feature must be treated with equal importance.

Train algo with normalization : Better accuracy

with no normalization : neglecting most of the features

(may miss out on imp info)

Why?

Transform data → similar distribution

→ & dimensionless

→ same units

EQUAL IMP TO ALL FEATURES

* Standardization, Z-score Normalization.

Transforming features in a way s.t

Mean (μ) = 0

Std dev (σ) = 1

$$x_{std} = \frac{x_i - \mu}{\sigma}$$

- SVM, K-means clustering → looking at NN ⇒ Belong to same cluster
  (usually scale our features)
  ⇒ If it is not of same units
  ∴ It may be not be the right neighbour.

4 features, 4 samples

∴ A → very large may only take this into consideration.

```
from sklearn.processing import standardScaler.
scaler = StandardScaler()
x_scaled = scaler.fit_transform(x)
```

mod value → similar line
(all datapoints lying in some range.)

* Min Max Normalization / Feature Scaling

→ converts feature values within a specific range
(mostly (0-1))

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

(NO OUTLIERS should be present)

∴ upper and lower bound may be influenced

Feature clipping (numpy.clip → all values in an array
0-100 → as is
above 100 → clips to 100.

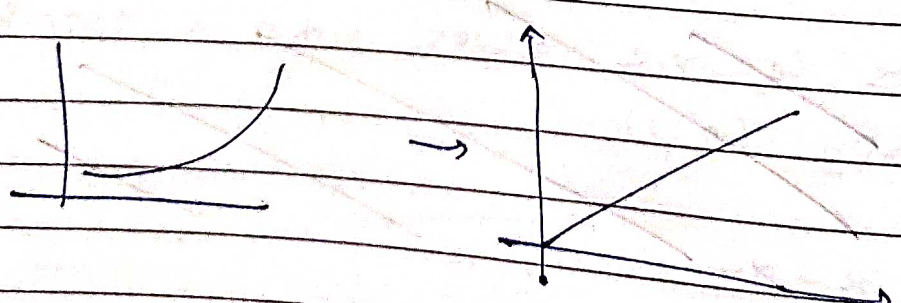
↳ defining lower and upper values

Ex: [10, 15, 200, 5, 25, 180]
↓
clip
↓
100

3 σ & limit — lower threshold = 0
higher threshold = 100

↳ Takes care of outliers, by limiting or bounding the extreme values of a feature with a specified range
↳ Determine clipping thresholds.
↳ Caps extreme values: The feature is constrained within a specific range.

* Log scaling
↳ When we deal with data that spans a wide range of values.
↳ It computes the log to compress a wide range to a narrow range, for a balanced visualization.



SUMMARY

Normalization Technique	Formula	when ?
Standardization	$x_{std} = \frac{x_i - \mu}{\sigma}$	when feature distribution does not contain extreme outliers.
Min-Max Scaling	$x_{nom} = \frac{x - x_{min}}{x_{max} - x_{min}}$	when the feature is one- or less uniformly distributed across a fixed range.
Feature Clipping	If $x > max$, $x' = max$ $x < min$, $x' = min$	when the feature contains some extreme outliers
Log scaling	$x' = \log(x)$	