feature engineering

- **Feature Transformation**
  - missing value imputation
  - Handling catego-rical features
  - outlier detection
  - feature scaling
- **Feature construction**
  - constructing new featurs manually
- **Feature Selection**
  - ✓ Selecting Imp featur
- **Feature extraction**
  - extrating new feature from the given features programatically but not manually

1.4 feature scaling

It is technique used to transform and standardize the independent features present in the data in a fixed range. of to a consistent and predefined range

- Standardization
- Normalization
  - → minmax scalar
  - → Robustness scalar
  - →

Standardization : Also called as Z-score Normalization. No specific range

For Ex :-

Age	Age'
$x_1 = 27$	x_1'
32	x_2'
64	x_3'
35	⋮
⋮	⋮
500 values	500 values

$$\boxed{x_1' = \frac{x_1 - \bar{x}}{\sigma}}$$

$\bar{x}$ = mean of age column

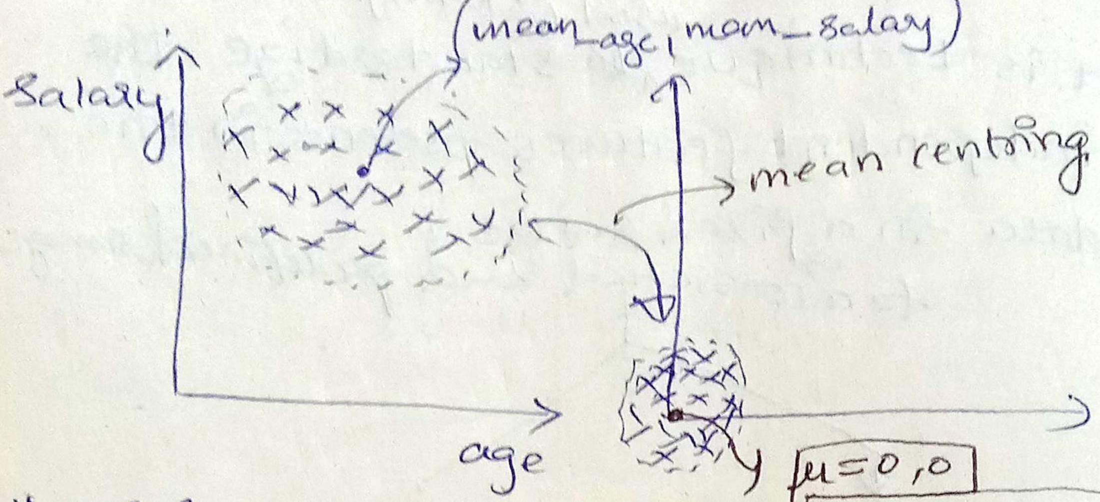
σ = standard deviation of age column

This column

$$\boxed{\begin{array}{c} \text{mean}(\mu) = 0 \\ \sigma = 1 \end{array}}$$

☆ ☆

intution:



$\boxed{\sigma_{age}, \sigma_{sal} = (1,1)}$

* If intially the $\sigma < 1 \Rightarrow$
  we will scale up the data
  so that ~~s~~ $\sigma = 1$, If $\sigma > 1$
* then~~s~~ we scale down it to $\boxed{\sigma = 1}$

☆ intution = mean centring + scaling by the factor of σ

* when you perform standardization on a particular column which has outliers. The outliers will not remove on standn. we should handle them explictly.

when to use standardization.

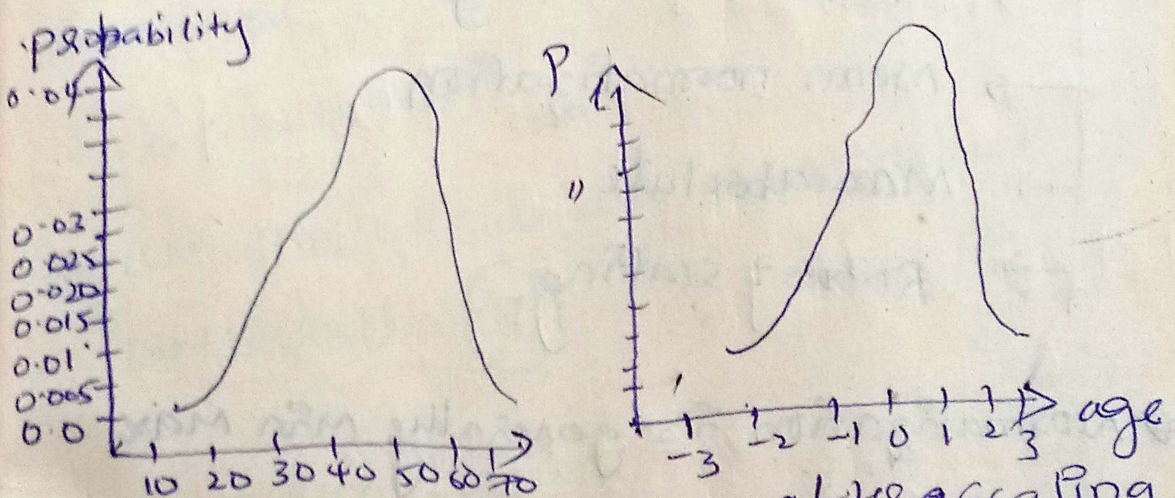
1. K-Means (")
2. K-Nearest neighbors (Euclidean distance)
3. principal component analysis
* (Try to get features with max variance)

4. ANN - because we apply gradient descent

5. Gradient descent -> During back propagation we update the weights to have similar scale weights we use.

* Distribution is always retained

=> Eg:- By using distplot of age



before scaling

after scaling

Standarization defn:-

Here all the features will be transformed in such a way that it will have the properties of a standard normal distribution with $\mu = 0$ & $\sigma = 1$ (standard devi)

Normalization :- It is a technique often applied as part of data preparation for machine L. The goal of normalization is to change the values of numeric columns to use a common scale, without distorting differences in the ranges of values or losing information.

- Min Max scaling
- Mean normalization
- Max absolute
- Robust scaling.

⇒ Normalization is generally Min Max scaling

* Min Max scaling :-

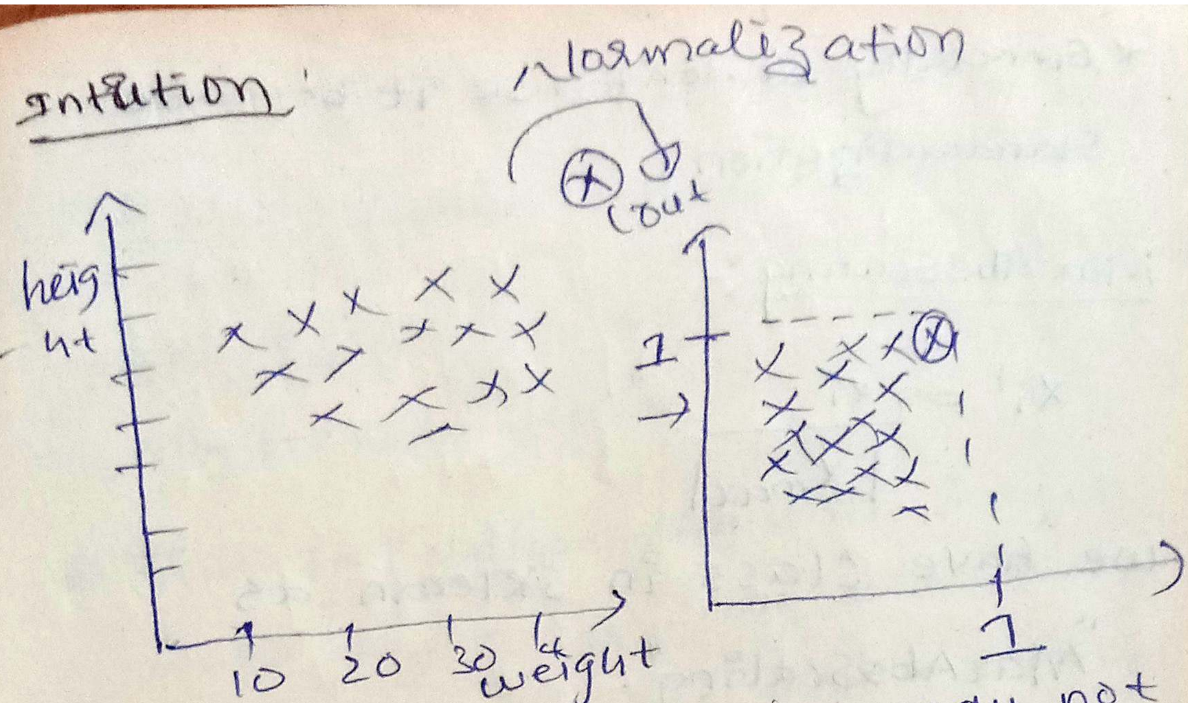
Age :-

$x_i =$ 21
17
16
34
56
⋮
100 values

$$\boxed{x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}}}$$

* Range of Nor :- [0 to 1]

Intution:



* After normalization distribution can may not be retained

* we should carefully think about outliers!! ?

Mean normalization:

$$x_i' = \frac{x_i - x_{mean}}{x_{max} - x_{min}}$$

Range = [-1 to 1]

Intution: Is to converge data toward ~~mean~~ centre through mean which is called as Mean centring

* Generally we don't use it because Standardization.

MaxAbsScaling :-

$$x_i' = \frac{x_i}{|x_{max}|}$$

we have class in sklearn as "MaxAbsScaling".

Use :- when we have sparse data.

Means data having more $\boxed{\text{Zero's}}$

Robust scaling :-

wt

x_i - 200

300

75

$$x_i' = \frac{x_i - x_{median}}{IQR}$$

IQR = Inter Quartile range

= (75^{th} percentile − 25^{th} percentile)

= $(Q_3 - Q_1)$

Use :- It is Robust to outliers

* Means when we have outliers we can use this

Normalization VS standardization

→ Q1: Is feature scaling required?

Q2: If yes, 90% we use standard scalar

only 10% normalscalar

* Tip!:- IN CNN we deal with images which rang 0-255 pixels we know the range then we use Normalization"

1.2 Handling categorical features:-

nominal features	ordinal features
⇒ No order of categories	The feature in which categories have order
Eg^n:- States	Eg:- Ed^n
Telangana	PG (post graduate)
maharastra	UG (under ")
Andhra.	HS (High school)
	order ⇒ PG > UG > HS
Sol^n: "One Hot Encoding"	Sol^n: 1) Ordinal Encoding
	2) label Encoding

Ordinal Encoding :-

1) **Label Encoding** :- we use this when our label/output column has categories like (yes/No) (classification)

Eg :-

No. of hours studied	No. of hours slept	pass/fail y
5	8	pass
6	9	fail
7	7	pass
8	6	fail

we use label Encoding

1
0
1
0

2) **Ordinal Encoding** :- we give order as PG=2, UG=1, HS=0

Education	(Edn)1
HS	0
PG	2
UG	1
PG	2
HS	0
UG	1

* Both label / ordinal encodings are same But label is only used on dependent feature, ordinal used on independent features

One Hot Encoding

Eg:- color

yellow

Blue

Red

Blue

yellow

Bule

SK-learn Pipelines :-

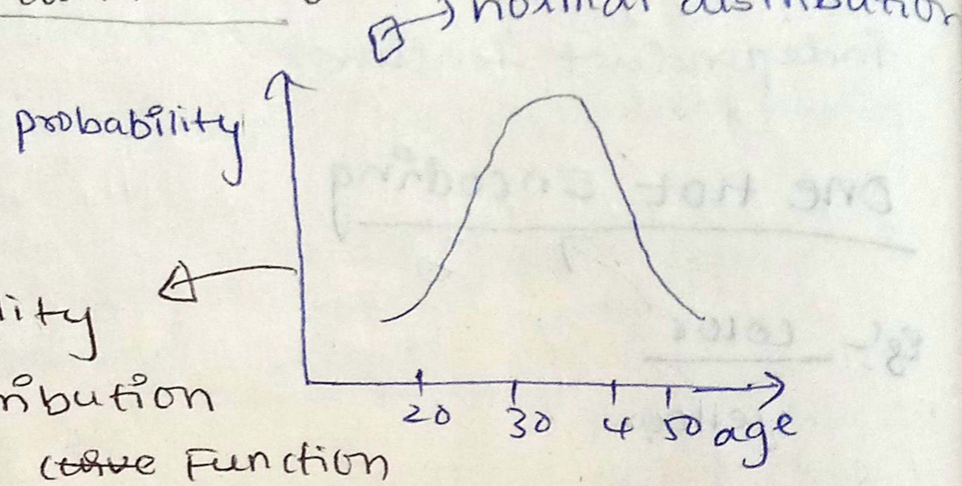
* pipelines is a mechanism chains together multiple steps so that the output of each step is used as input to the next step.

* pipeline makes it easy to apply the same preprocessing to train & test

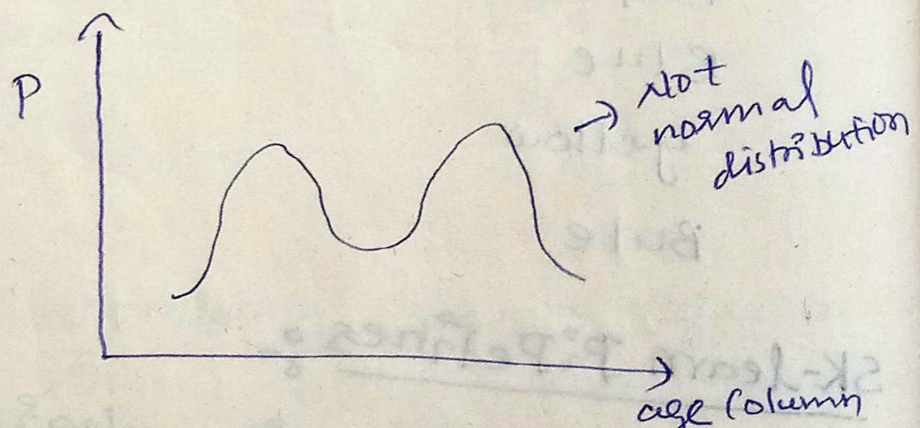
Function Transformers :-

Goal :- Is to make ~~son~~ normal distributio

normal distribution :



probability distribution ~~curve~~ Function (PDF)



Use of normal distribution :- In ML we have statistical algorithms there are linear regression, logistic regression ... etc. so that we can get good accuracy.

Sklearn Transformers

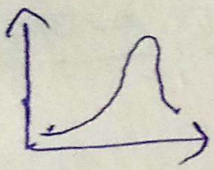
- Function Transformers
  - log trans
  - Reciprocal trans
  - sqrt trans
  - custom (we can make)
- power trans formes
  - Box-cox
  - yeo-johnson
- Quantil trans

1) How you find the normal distribution?

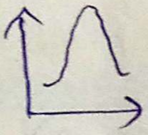
i) sns.distplot

ii) pd.skew()

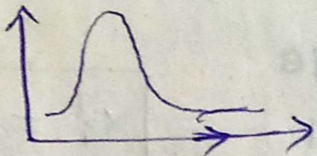
- -1 negatively skewed



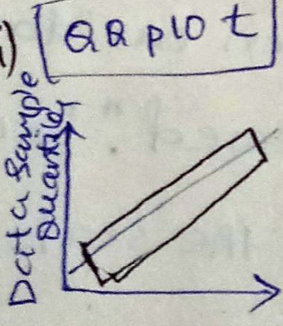
- 0 normal

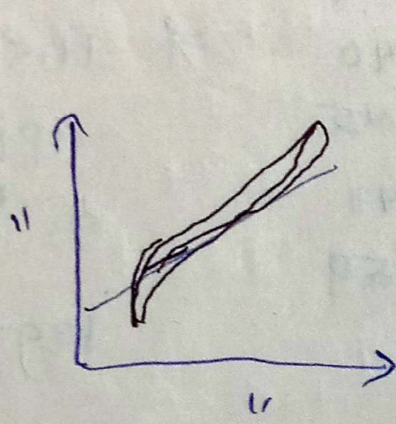


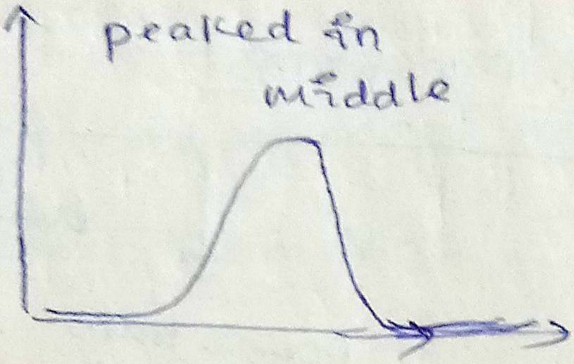
- +1 postively skewed



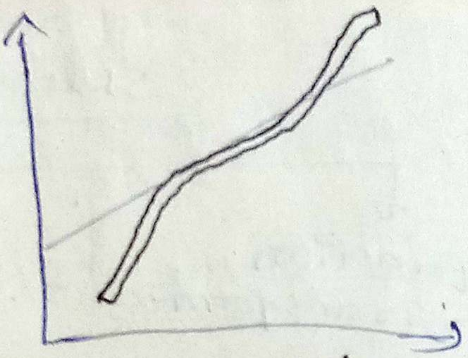
iii) $\boxed{\text{QQ plot}}$



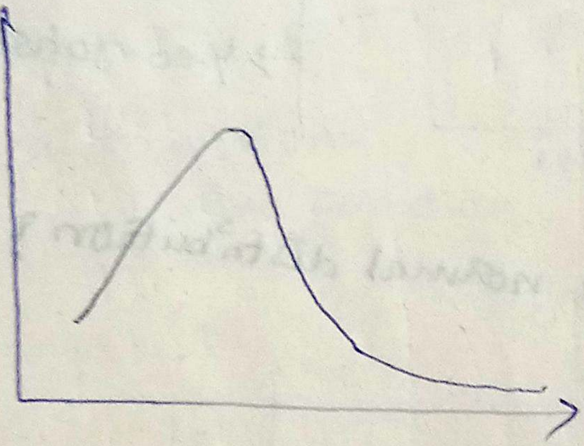




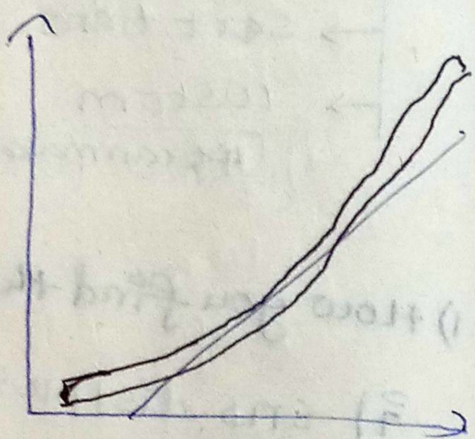
distpolt



QQ-plot



+ve skewed



QQplot

log transformation

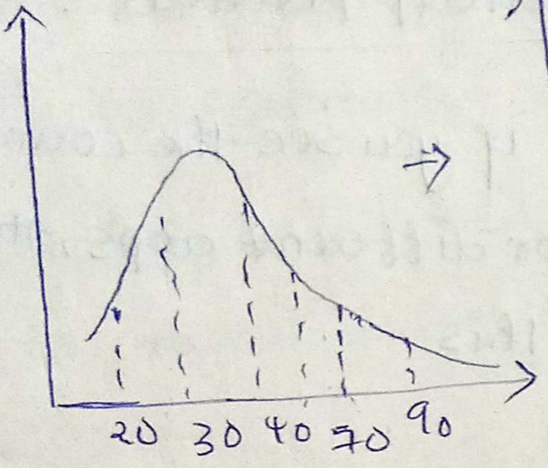
eg: Age

$x_i = 21$
35
40
45
47
50

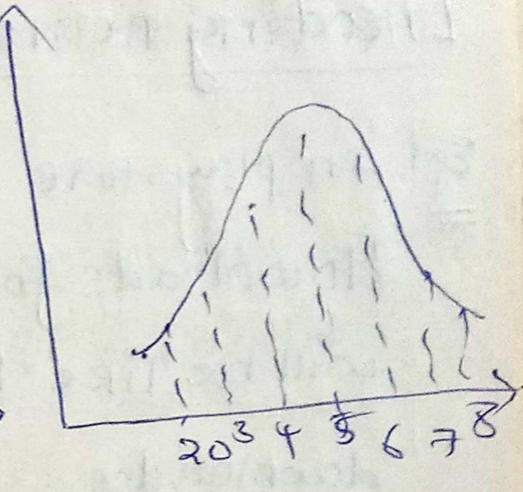
$$\boxed{x_i' = \log(x_i)}$$

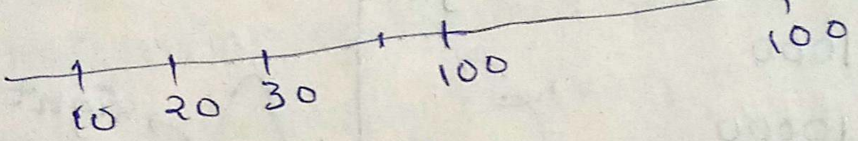
* The log transfoom is applied when our data is "+vely skewed". Because log decrease the scale.

eg:-

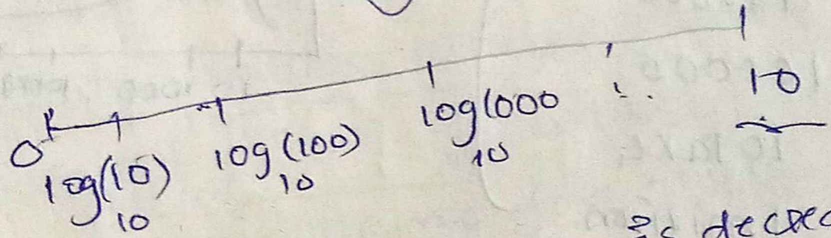


→





⇓ log



⇒ range is decreased

Reciprocal: $(\frac{1}{x})$: square (x^2) : sqrt $(\sqrt{x})$

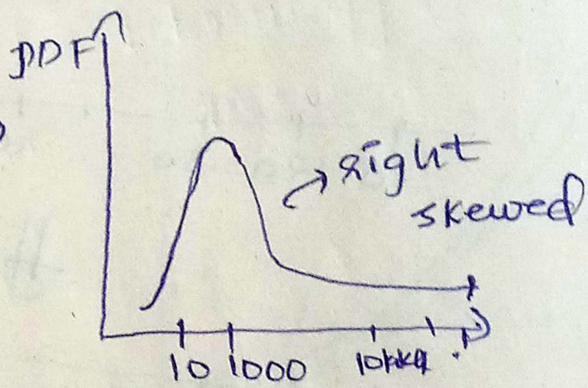
used for left sekwed data

Encoding numerical Features :-

Eg: In playstore if you see the count of downloads for different apps, they will be like this:

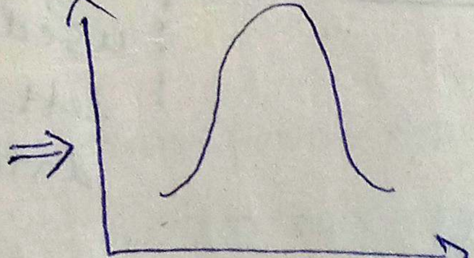
downloads

27
1000
10000
950000
100000
10 lakh
1 million



But when you convert them into certain ranges (or) bins like this:

10 – 1000
1K – 10K
10K – 100K
100K – 1000K
1000K – 10000K



* what we have done above ?

we have converted the "numerical data" to "categorical data".

* The technique we use to convert numeri-cal to categorical is "Discretization".

1) Discretization :- It is a process of transforming continuous variables into discrete variables by creating a set of contiguous intervals that span the range of variable's values. Discretization is also called $\boxed{\text{binning}}$, where bin is an alternative name for interval.

Why use Discretization :-

1. To handle outliers
2. To improve the value spread.

Types of Discretization :-

Binning

- unsupervised
- supervised
- custom binning

unsupervised — Supervised — custom binning

unsupervised:
- Equal width (uniform)
- Equal frequency (quantile)
- kmeans binning

Supervised:
- Decision Tree Binning

Equal width / uniform Binning :-

Age :- 27, 32, 84, 52, 61 160

$max_{age} = 160$

$min_{age} = 0$

we select the no. of bins = 16

$$\Delta \text{Range} = \frac{max\_age - min\_age}{\text{bins}}$$

$$= \frac{160 - 0}{16} = 10$$

Now bins range will be

(0 - 10), (10 - 20), (20 - 30), ... (150 - 160)

Total = 16 bins

Use: outliers are also present in last (or) any certain range.

ii) No change spread (or) distribution.

Equal Frequency / Quantile Binning :-

⇒ we should decide no.of intervals = 10

Each interval contains 10% of total observations.

0-16; 16-20; 20-22; 22-25; - - - -

50-74

Note! Here bin width is not constant

K-Means Binning :-

Used: when our column is in the form of clusters (or) groups

Step-1

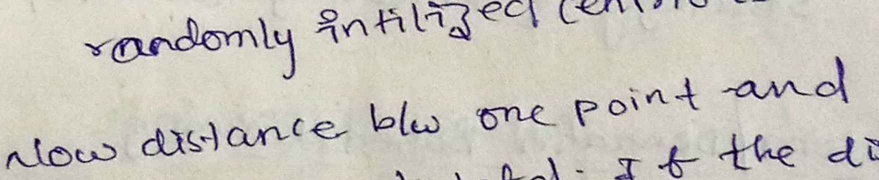
00000 ⟷ 00000 ⟷ 0000 ⟶

14 15 16 17 18   21 22 23 24   26 27 28 29   ''

18   22   24   age column

randomly initilized centriods

Step-2 Now distance b/w one point and every centroid is calculated. If the distance is less then that the point assigned to a particular centriod.



Step-3 Now mean of each point is selected in cluster is taken then centriod position is changed within the cluster.

Step 4 :- process repeated when position of centriod is changed.

* finally the range of each bin is

$0 - c_1$; $c_1 - c_2$, $c_2 - c_3$ ...

Code :

from sklearn.preprocessing import KBinsDiscretizer

parameters :-

n_bins = our wish (any int)

encode = {'onehot', 'one-hot-dense', 'ordinal'}

strategy : {'uniform', 'quantile', 'kmeans'}

Custom / Domain Based Binning :-

* we create own range like

[0-18] → kids

{18-60] → 2nd range

(60-80] → 3rd range

* This class is not present in sklearn. we should use

pandas.

2. Binarization:

For ex:-

age	age'
16	0
17	0
12	0
22	0
35	0
45	0
77	1
65	1
62	1
54	1
80	1

I set a threshold = 45

If my age ≤ 45 then

If $age_i \le 45$ $age_i' = 0$

$age_i > 45$ $age_i' = 1$

Correlation vs Co-variance

Correlation	Co-variance
Direction ✓	Direction ✓
Strength ✓	Strength ✗
Range = -1 to 1	Range:- $-\infty$ to $+\infty$

Mixed data:

For ex:

Type-1:

cabin	categorical	Numerical
B5	B	5
D4	D	4
S2	S	2
H3	H	3
C23	C	23
D41	D	41

Type-2:

column	cat	num
7	7	NaN(null)
3	3	NaN
1	1	NaN
A	NaN	A
C	NaN	C
4	4	NaN
3	3	NaN

* MISING Values

- Remove the rows
- Impute
  - univariate
    - numerical
      - mean/median
      - Any random value
      - End of distribution
    - categorical
      - most frequent value
      - missing
  - multivariate
    - KNN Imputer
    - Iterative imputer

Complete case Analysis (CCA):-

(or)

* Complete case analysis (CCA), is also called "list-wise deletion" of cases, consists in discarding observations where values in any of the variables are missing.

* Complete case analysis means literally analyzing only those observations for which there is information in all of the variables in the dataset

Assumption for CCA:-

we apply the CCA when our data is randomly missing

* we don't apply CCA:-

  i) Let us assume in one of the column if 1st 50 rows values or last 50 rows values are missing then we don't apply it

* we follow $\boxed{MCAR}$ = missing complete at random.

* df.isnull().mean()*100

we get percentage of missivalues in each column

Numerical data:-

1) * Mean/Median is discussed before.

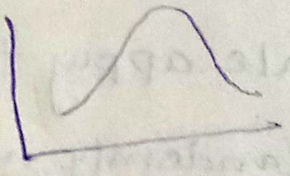
2) * Arbitary value constant

SimpleImputer(strategy='constant', fill-value = any value)

when to use:- When dat is not missing at random.

3) End of Distribution Imputation:-

when to use:- When data is not missing at random.



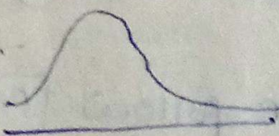
Case-1

* If normality distributed

the we fill with (mean + 3σ) ✓
(or)
mean − 3σ ✓

Case-2

If +vely or −vely skewed then we use



* we fill with $Q_1 - 1.5\,IQR$

$Q_3 + 1.5\,IQR$

$Q_1 = 25^{th}$ percentile value

$Q_3 = 75^{th}$ percentile value

$\boxed{IQR = Q_3 - Q_1}$

Disadvantges

→ outliers

→ Distribution (pdf)

→ covariance change

→ variance pes (or) des largely

univariate-categorical

* most frequent:

codn: If data is MCAR

ii) 5<%

* "Missing": Here we will with the word 'Missing' in place of Nan values

codn: Not MCAR

ii) 5>%

Random Imputation :-

* It is applied on both numerical and categorical data.

* It doesn't matter
  - i) MCAR
  - ii) <5%

* Eg

Sex	Age
M	26
F	32
	54
M	—
F	29
—	12
M	—
F	52

Randomly chosen

These missing values will be replaced by the random number in that column

* Imp:-
  - preserves the variance of the variable
  - Memory heavy for deployment, as we need to store the original training set to extract values from and replace the NA in coming observations
  - Well suited for linear model as its does not distort the distribution, regardless of the % of NA

KNN Imputer

	col1	col2	col3	col4
R1	12	14	-	13
R2	$\boxed{A1}$	15	16	17
R3	14	12	17	25

focussed

(12, 14, -, 13) = point 1

point 2 = (-, 15, 16, 17)

* The distance b/w (1, 2, 3) & (7, 8, 9) is calculated through "Euclidean distance"
* But we have missing values so we use "nan_euclidean"

nan_euclidean b/w P1 & P2

For ex: nan_euclidean

$$= \sqrt{\text{weight} \times [(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \ldots]}$$

$$\text{weigh} = \frac{\text{Total no. of actual pairs}}{\text{no. of pairs present}}$$

R1 & R2

$$\text{dist } = \sqrt{\text{weigh} \times [(14-15)^2 + (13-17)^2]}$$

$$R_1 \& R_2 = \frac{3}{2} \times [(\ ) + (\ )] = 10$$

$R_2 * R_3 = \sqrt{\frac{3}{3} \times [(\ \ )^2 + (\ \ )^2 + (\ \ )^2]} = 50$

* If no. of neighbors = 2 = K

The $A_1 = \frac{12+14}{2}$

weights = uniform ⇒ we have take the mean of the nearest rows.

If weights = 'distance'

then $A_1 = \dfrac{\frac{1}{d_1}(x_1) + \frac{1}{d_2}(x_2)}{2}$

$= \dfrac{\frac{1}{10}(12) + \frac{1}{50}(14)}{2}$

= —,,

Advantages:

i) most accurate values than mean/ median/ random

Disadv:

→ If the dataset is large then it takes so much time to fit. means be-cause it should calculate more distances.

ii) During the deployment time we should upload our training dataset. because it any input value of the user is missing we should calculate the distances.

Iterative Imputer / MICE :-

MICE = Multivariate Imputation by Chained Equations.

Assumptions

MCAR = Missing completely at random
MAR = " at random
MNAR = " Not " "

* we get Good results of mice when $\boxed{MAR}$ cod^n is present

Adv & Dis adv

Adv ↓ most accurate

Dis adv ↓
i) on large dataset its working is very slow
ii) memory = we should upload the data set on the server.

Outliers:- The data that behave different -ly from others.

→ Big Question!:- In some situations outlier can also be use & dangerous, when should them.

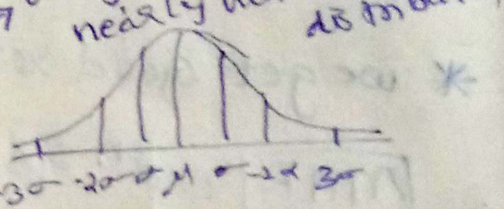
Effect of outliers on ML algorithms:-

* The algorithms in which the weights are assigned are generally effected. They are:

i) Linear Regression
ii) Logistic Regression
iii) Adaboost
iv) Deep learning

How to $\boxed{\text{detect}}$ outliers:

1. Normal distribution → only if our data is nearly normal distributed



For ex! age col^n

* If data in age col is m
> μ+3σ OR < μ-3σ then it is outlier.

2. Skewed Distribution:

IQR = Interquartile Range