

# AML

Part-1 ↪ ↪ Part-2 ✓

(90%)

↪ Recommender Systems

↪ Time-Series Analysis

## \* Recommender Systems:

Netflix

Youtube

Amazon

Laptop

→ Added to Cart

Instagram

Bumble

Frequently Bought Together

'Keyboard' 'Mouse'

(Recommendations)

- R.S. predicts what a user might be interested in by :

[FRIENDS] ✓

[Jolly LLB - 3] ✓



Comedy

Movies / Web-Series

① past purchasing / browsing history

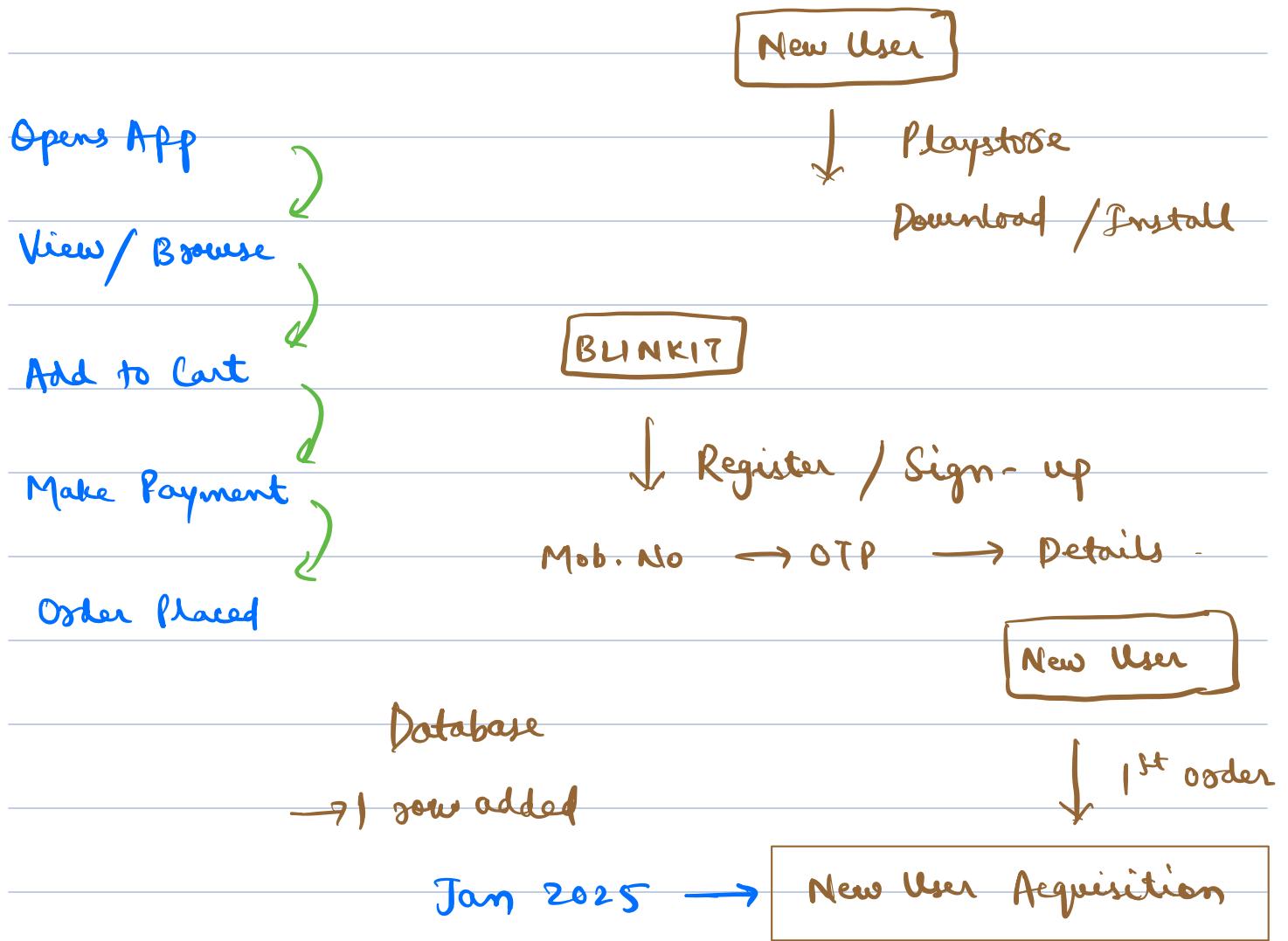
② Similar users behaviour.

### ③ Similar products.

Why companies need R.S. ?

- ① Improve User Experience
- ② Increase Revenue / Profit
- ③ Reduce Search Time

USER  
ACTIVITY - FUNNEL :



Feb 2025

Feb 2025

Place 2<sup>nd</sup> order

Place NO order

→ Retained User  
(Retention)

→ CHURN

March 2025

2<sup>nd</sup> order [Re-activated Customer]

## \* Types of Recommender Systems:

### ① Popularity-based:

Trending on Insta → SPOTIFY

- Recommends 'most bought' or 'most viewed' item.
- No personalization
- Used for new users.

### ② Content-based filtering:

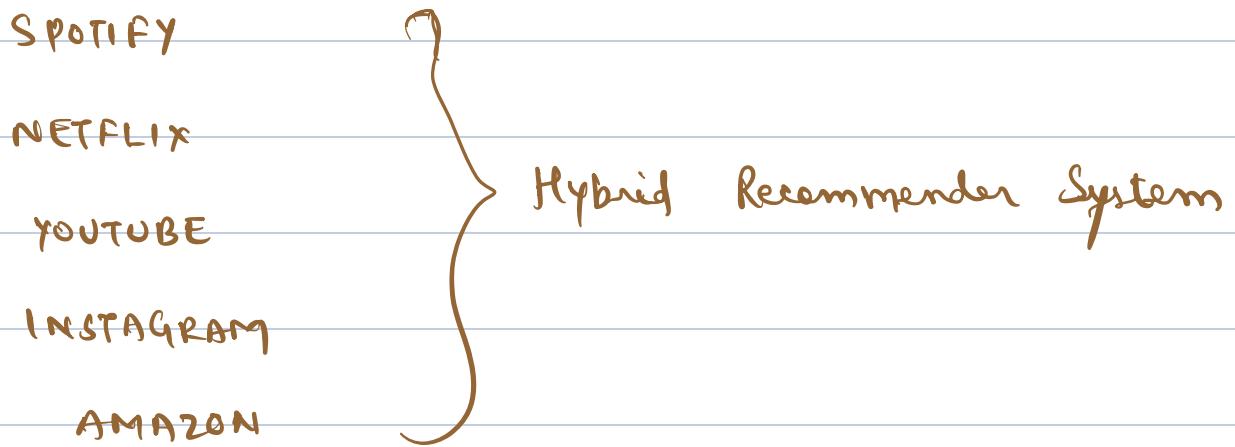
- Recommends items / products as per what user liked / watched earlier.

Ex: Comedy shows → Comedy MOVIES

### ③ Collaborative Filtering:

- People like you bought this / watched this.

- Learns from user behaviour, ratings, purchase history



- Market - Basket Analysis

↳ Improves:

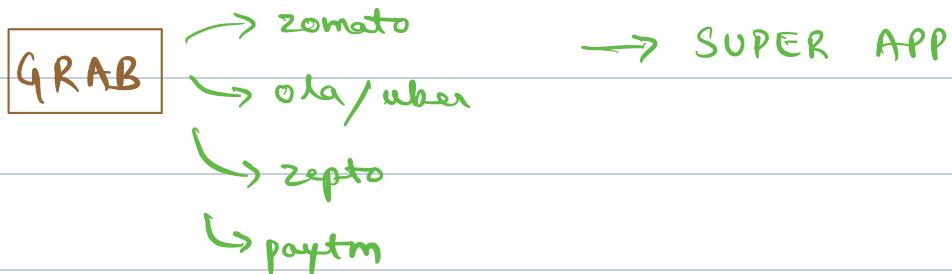
UPSELL :

① Product Placement / Inventory

CROSS-SELL :

② Combo offers and bundles

③ Cross-selling recommendations



- upsell**
- user → 100 Grabtaxi rides → 200 G-T. rides
- cross-sell**
- user → 100 Grabtaxi rides → 5 GRAB food orders

60 Rs/-

onion - 30/-

tomato - 20/-

potato - 10/-

[Market - Basket Analysis]

combo (onion, tomato, potato)

- 55/-

## → APRIORI ALGORITHM :

Core Idea :

(Milk, Bread, Butter)

Transaction ID

Items Purchased

1

Milk, Bread, Eggs -

2

Bread, Butter

3

Milk, Bread, Butter -

4

Bread, Eggs

5

Milk, Bread, Butter, Cookies -

6

Milk, Cookies -

7

Bread, Butter, Eggs

8

Milk, Bread -

9

Bread, Cookies

10

Milk, Butter, Bread -

## Steps :

- ① Calculates Support for each item.
- ② Form 2-item combination and calculate support again.
- ③ Generate few Rules

• Confidence

• Lift

$> 1$  (Strongly Associated)

$\leq 1$  (Weakly Associated)

$$\text{Support} = \frac{\text{Count of item}}{\text{Transactions count}}$$

### ① Calculate Support

Item	Count	Support
Milk	6	0.6 ✓
Bread	9	0.9 ✓
Butter	0.5	0.5 ✓
Eggs	4	0.4 ✓
Cookies	3	0.3 ✓
X Curd	2	0.2 ✗

Assume min. support as 0.3 (30%).

### ② Form 2-item sets & calculate support

Milk, Bread	5	0.5 ✓
Milk, Butter	3	0.3 ✓
Milk, Eggs	2	0.2 ✗
Bread, Butter	5	0.5 ✓

Bread, Eggs

2

0.2 X

$$D = \{1, 2, 3, 4, 5, \dots, n\}$$

$$\text{No. of subsets} = 2^n$$

③ Confidence. ( $X \rightarrow Y$ )

Support (Milk, Bread)

Support (Milk)

Confidence  $\approx (0, 1)$

$$\text{Confidence } (X \rightarrow Y) = \frac{\text{Support } (X \cap Y)}{\text{Support } (X)}$$

$$= \frac{0.5}{0.6} \approx 0.83$$

④ LIFT

$$\text{Lift} = \frac{\text{Confidence } (X \rightarrow Y)}{\text{Support } (Y)}$$

$$= \frac{0.83}{0.9} < 1 \quad (\text{Weakly Associated})$$

②  $(\text{Bread}, \text{Butter}) \rightarrow 0.5$

$(\text{Butter}) \rightarrow 0.5$

$$\text{Confidence} = \frac{0.5}{0.5} \rightleftharpoons 1.$$

$$\text{Lift} = \frac{\text{Confidence}}{\text{Support}(\text{Bread})} = \frac{1}{0.9} > 1$$

(Strongly Associated)

Apriori Algorithm Output tells -

① Frequent itemsets

② Association Rules ( $A \rightarrow B$ )

Limitations :

① Slow with large datasets

② Not suitable with continuous data

• ASSOCIATION RULES :

↳ finding relations between frequent item- sets

Notation :  $X \rightarrow Y$

→ People who buy 'X' have a very high likelihood to buy 'Y'.

Ex:  $X \rightarrow \{ \text{milk, bread} \}$

$Y \rightarrow \{ \text{jam, eggs} \}$

$\{ \text{milk, bread} \} \rightarrow \{ \text{jam, eggs} \}$

(Antecedent)

(Consequent)

if

then

Pizza  $\rightarrow$  Coke  $\gamma$

$X \rightarrow$  garlic bread  $\gamma$

## FORMULAS

$$\textcircled{1} \quad \text{Support}(X) = \frac{\text{No. of transactions in which } X \text{ is present}}{\text{Total transactions}}$$

$$\text{Support} = \frac{x}{N}$$

## ② Confidence

$X \rightarrow Y ?$

$$\text{Confidence } (X \rightarrow Y) = \frac{\text{Support } (X \text{ and } Y)}{\text{Support } (X)}$$

$$= P(Y | X) \text{ (logically)}$$

### ③ Lift :

$$\begin{aligned}\text{lift } (x \rightarrow y) &= \frac{\text{Confidence } (x \rightarrow y)}{\text{Support } (y)} \\ &= \frac{P(x \text{ and } y)}{P(x) \cdot P(y)} \quad (\text{intuitively})\end{aligned}$$

- $\text{lift } (x \rightarrow y) = 1$ ,  $x$  and  $y$  are independent
- $\text{lift } (x \rightarrow y) < 1$ , unlikely to be bought together
- $\text{lift } (x \rightarrow y) > 1$ , likely to be bought together

Lift varies from  $(0, \infty)$

Original Data:

Invoice	Stock Code	Description	Quantity
---------	------------	-------------	----------

↓ To prepare data for  
Apriori algorithm

① Filtered UK Data

② Grouping on invoice No. and Description → Quantity Sum

Transaction 1

Mug 6 ←

Transaction 1

Mug 4 ←

[Invoice, Description]. quantity.sum()

→ unstack()

invoices	product	total quantity
1	Milk	6
1	Bread	10
1	Butter	2

unstack()

↓ pivot table (rows → columns)

invoices	Milk	Bread	Butter
1	6	10	0

"Each product becomes a separate column"



Invoice No

Milk

Bread

Butter

|

|

|

0

→ Association Rules :

→ Co-occurrence

If any transaction / invoice have

1 single item, remove those

→ Invoices → at least 2 distinct items are present