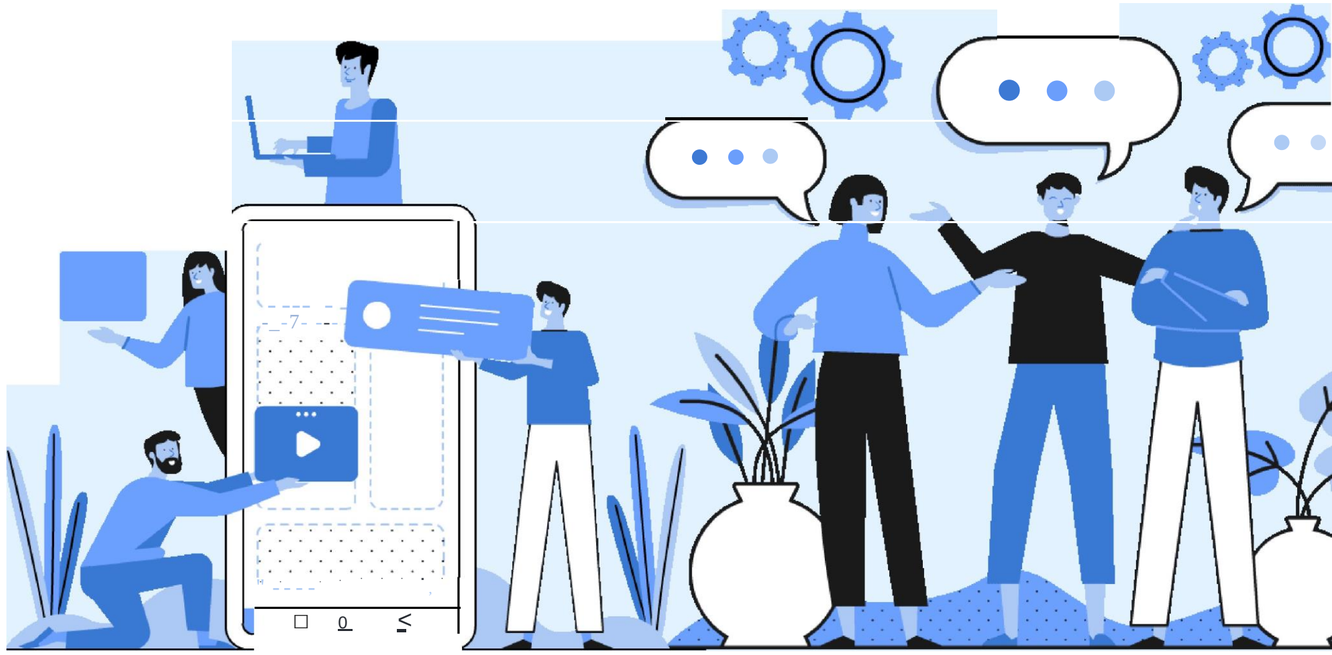
Affinity analysis using ML

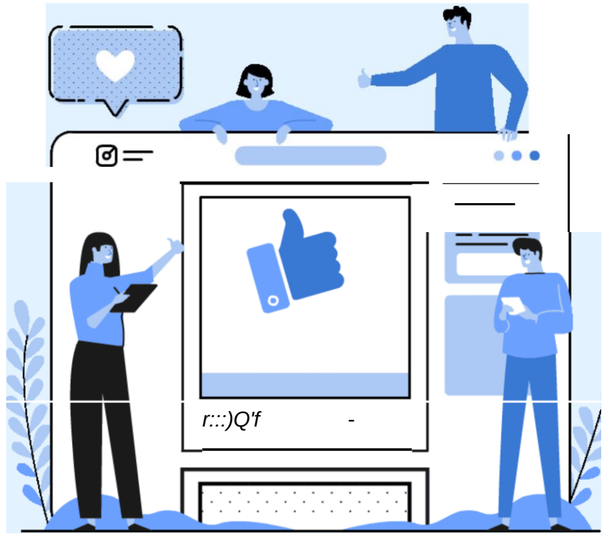


Hello & welcome to presentation

Presented by:-
Prem Shankar Pandey

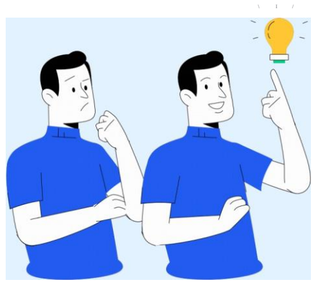
Table of

Contents



1.

Overview



Abstract

- Affinity analysis is basically the way to find the patterns to determine connections between purchases in order that stores can increase their cross-selling potential.

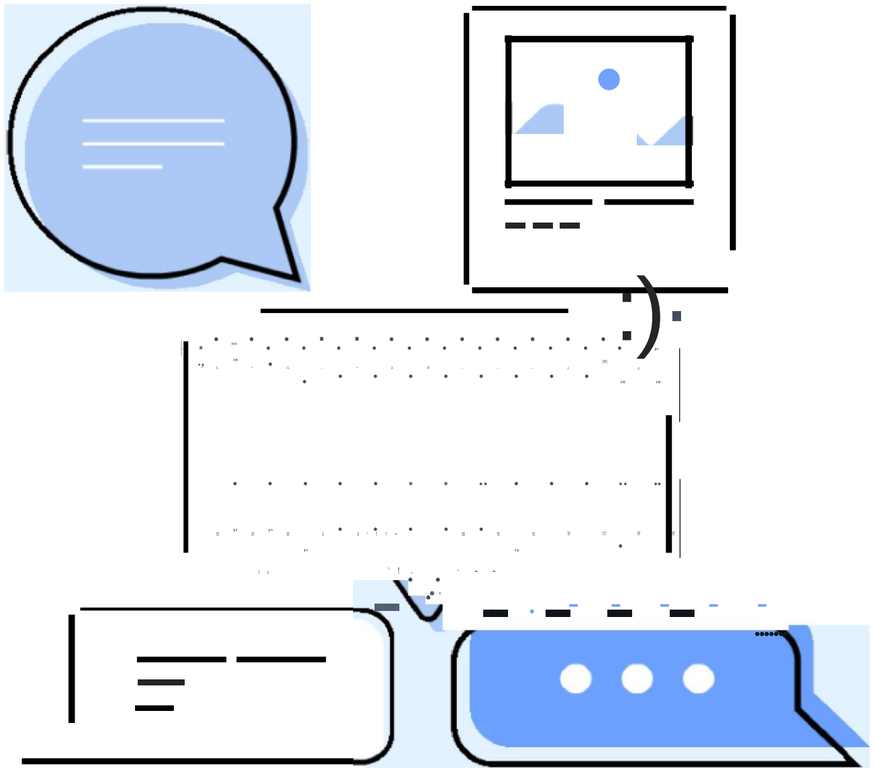
- It can help us to predict what the customer is going to buy next by looking at the products he is buying.

Introduction

Apriori is a well-known algorithm used in data mining to identify frequent item sets and association rules in transactional data. It uses a "bottom up" approach to generate frequent item sets, where the algorithm starts with individual items and combines them into larger and larger item sets as long as the item sets meet a minimum support threshold.

In the context of affinity analysis, the Apriori algorithm is used to discover associations between items in transactions. For example, it can be used to find association rules between items purchased in a grocery store. The algorithm would identify items that frequently occur together in the same transaction and generate rules to describe the relationship between the items.

The resulting association rules can be used to make predictions about future customer behavior, such as recommending items that are often purchased together. They can also be used to inform product placement and pricing decisions, and to improve the overall customer experience.



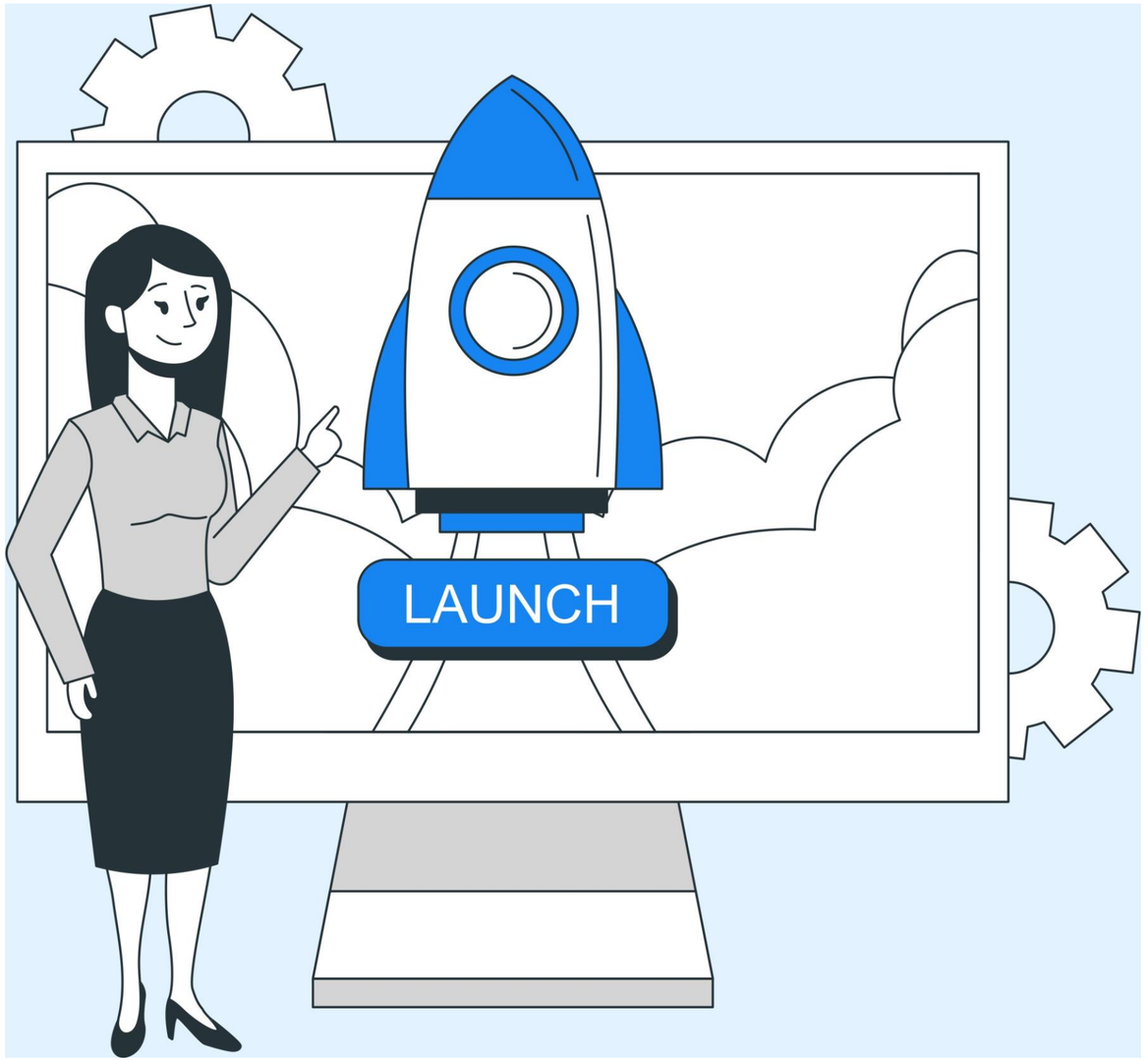
Problem statement

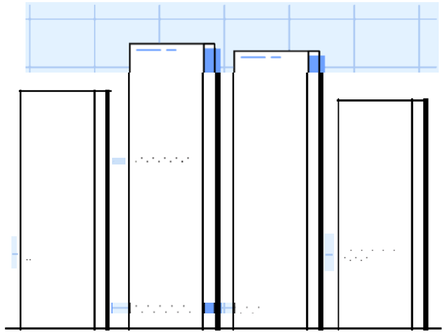
We tend to deliver a recommendation in this project that can be employed as an effective technique to generate the frequent itemset and assist us in identifying the most important itemset. As a result, the Apriori Algorithm will assist us in identifying the most irritating and necessary itemset, which will improve market strategies, customer convenience, and boost the number of goods sales, resulting in a lucrative business.

2

Methodology

We will be discussing the basis, the approach, the working of algorithm and finally to implement our project





Basis

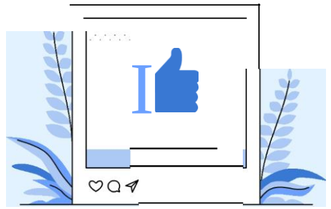
It is based upon the theory that if you buy a certain group of items, you are more (or less) likely to buy another group of items.

Beer-chips- Meal

if you are in a pub and you buy a pint of beer and don't buy a bar meal, you are more likely to buy crisps at the same time than somebody who didn't buy beer.

Cigarette-Chewing gum-Sweet

If a smoker buys a cigarette, then there is high chance that he will buy a chewing gum rather than eating something sweet. This creates a relation



A Picture is worth a thousand words

So, I did some research and found something.

Literature review

Item-Basket Revenue Maximization

They conclude that a greedy static ordering of nodes based on potential revenue maximization leads to higher overall revenue amongst the local search heuristics investigated. If prices of all items are more or less similar, then greedily picking nodes based on their influences and locally fixing their prices would lead to a good overall revenue gain.

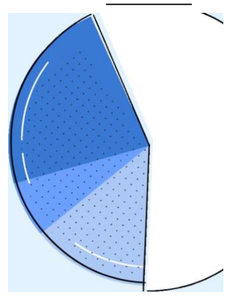
Market Basket Analysis: A Profit Based Product Promotion Forecasting

This paper deals with the data mining using Eclat Algorithm to figure out the consumer buying behavior patterns, which reveals the frequent sales and non-frequent sales in products. The prediction is done by the aid of mined dataset from the sales department of the retail shop. By generating the frequent itemset it is easier to identify the most necessary itemset and the most annoying itemset.

The Analysis and Prediction of Customer Review Rating Using Opinion Mining

To analyze the review system of customer feedback. They tend to improve the customer feedback system as it has some flaws due to lack of understanding between the software and the customer. The rating read by customer and the rating summarized by the algorithm may confuse the customer

Approach



We are utilizing the **Apriori approach,** which posits that any subset of a frequent itemset must also be frequent.

We chose this model since it allows us to discover such patterns inside goods in a shorter amount of time

Apriori Algorithm

For finding frequent itemsets in a dataset for boolean association rule. It is called so because it uses prior knowledge of frequent itemset properties.

Step 1

- Set a minimum support threshold (e.g., if an itemset occurs in more than X% of transactions, it will be considered frequent).
- Extract all the individual items from the transactions and count their frequency of occurrence.

Step 2

- Compile a list of frequent items (i.e., items that meet the minimum support threshold).
- Generate all possible combinations of frequent items to form candidate itemsets.

Step 3

- Calculate the support for each candidate itemset by counting the number of transactions in which it occurs.
- Prune any candidate itemsets that do not meet the minimum support threshold.

Step4

- Repeat steps 2-1 to 3-2 for each size of itemsets, starting with 2-itemsets and extending to larger itemsets.
- Generate the set of frequent itemsets, which represent frequent patterns in the data.

Step 5

- Use these frequent itemsets to generate association rules, which describe relationships between items.
- Analyze the generated rules to identify the most interesting and relevant patterns in the data.

TID	ITEMS
T1	134
T2	2_,3,,5
T3	]l.,.2 3,S
4	.275
TS	1_,3,5

Assumi1n_g Mi1ni'imum Sup,p,ort Count o be 2

Item Set	Support
{1}	3
{2}	3
{3}	4
{4}	1
{5}	4

H,ere item ,**4 "s ,e "minated**

Item Set	Support
(1,2}	1
(1,3)	3
(1,5}	2
(2,3)	2
(2,5}	3
(3,5)	3

{1,2} is disca1rded

Item Set	Support
{1.,2,3}	**O**
{1,2,5)	**NO**
{1.,3,5}	2
{,2_,.3,5}	2

Pruning technique is applied

Item Set	Subset	Support
{l_,.2,3}	(1, **3}, {l, 2}, {2,** 3}	**NO**
{1,2,5}	{1, **2}, {1, 5}, {2, 5)**	**NO**
{1.,3,5}	(1, **3}, {1,** 5}, **{3, 5)**	2
{2,3,5J	**(2, 3}, {2., 5}**, {3, Si)	2

Item Set	Support
{l_,.3,5}	2
{2,3,5)	2.

Item Set	Support
{1.,2,3,5)	1

Final Table

Item Set	Support
{1.,3,5}	2
{2,3,5}	2

Let {1,3,5} and {2,3,5} be I and its subsets be S.

We generrate the ru1le:

For every Subsets S of I,

.S-> 1-S) Le. 1(Sr ecomm ends, I-S)1

If confidenoe>=min confidence value

Example:

.Setting 1m11in_confiid ence va lu e as 60%

(1,3,5,}

Ruilel: {1,3} -> ({1,3,5} - {I,3}) me ans I & 3 reco1mmends 5

Calcu lating Confidence: support (1, 3, 5)1/ supp,ort (1,3) = 2/3 = i66.66%>60% (Rule Accepted)

(5}

Ruile2: {S} > ({I,3,S} (5}) means S recommends and 3

Confidence: suppi0rt (1,3,5) / support 1(5,)=2/4=50%<60% (Rule Decrned)

Implementation

- For this analysis, we are using datasets from several countries. The datasets are organized in a list format, with the transaction id as the first element and a list of item ids for products purchased together as the second element.

- Items is a list of the items' names. The entire number of items in the dataset is stored in the no of items field. These records comprise tuples with the itemset as the first element and a collection of transaction ids containing the associated itemset as the second element.

- We can easily acquire the output dataset of the items that are usually bought along with a given item in stores for a specific country after applying the Apriori algorithm.

- As a consequence, it distinguishes between the most aggravating and necessary things in a certain store. As a result, market strategies, customer convenience, and the number of items sold are all improved

Steps

We will be doing a set of experiments on two data sets to evaluate the performance of the Apriori algorithm: a Market Basket Optimization available from the basket optimization repository and genuine e-commerce data with over 50,000 records. These data sets were chosen because they are commonly used standard benchmark data sets

- Association rules analysis is a technique to uncover how items are associated to each other. There are three common ways to measure association

Stat 1

Measure l: Support.
This indicates how popular an itemset is, based on the number of times it appears in total transactions. In other words, we say item frequency.

Stat 2

Measure 2: Confidence.
This indicates the likelihood of purchasing item Y when item Xis purchased, given as {X -> Y}. This is determined by the percentage of transactions in which item X appears with item Y.

Stat 3

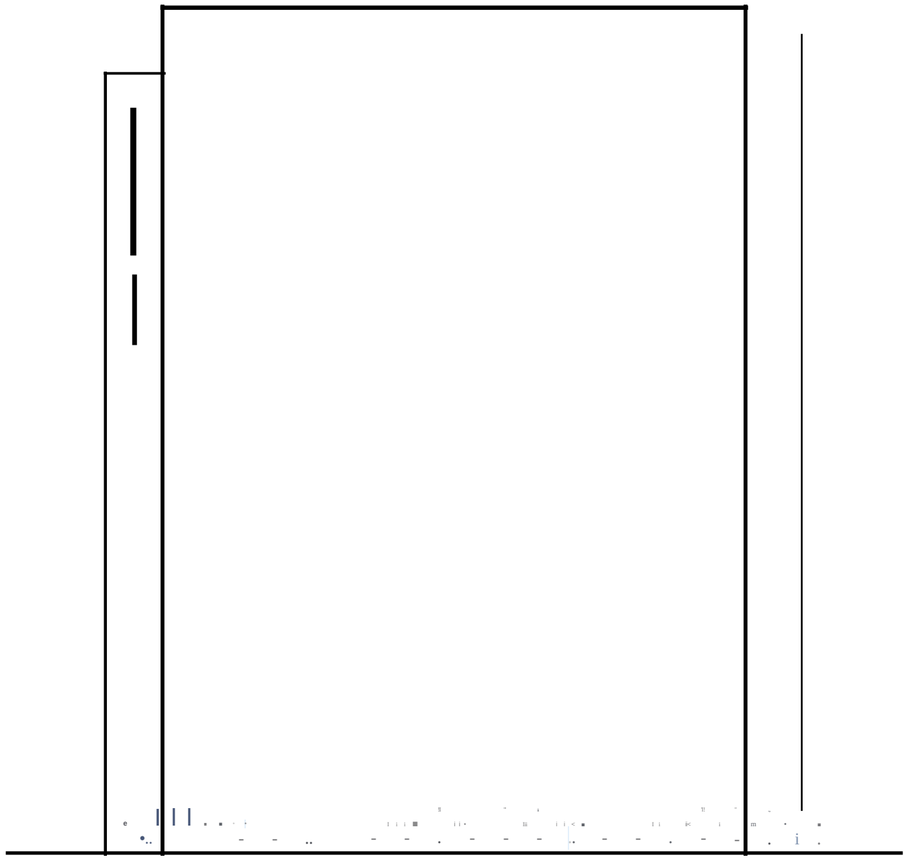
Measure 3: Lift.
It's the proportion of expected to observed confidence. It is defined as the ratio of Y's confidence when item X was already known(x/y) to Y's confidence when item Xis unknown. In other words, Y's confidence in relation to x and Y's confidence in the absence of X. (means both are independent to each other).

Formulas defined

Support = Occurenee / total no of trans

Confidence= Support(X union Y) / Support X

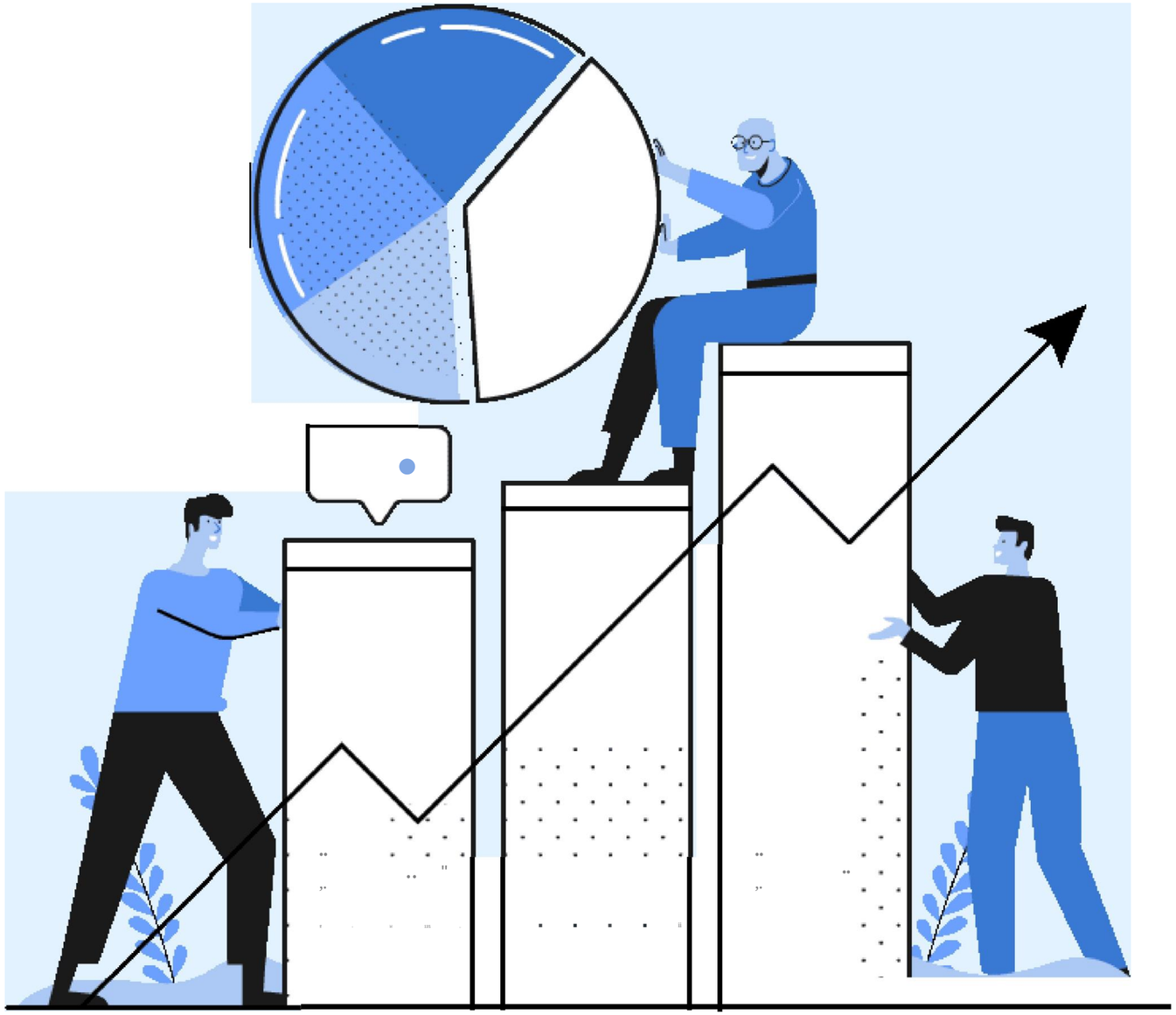
Lift= Support(X Union Y)/ Support(X) * support(Y)



3.

Outcomes

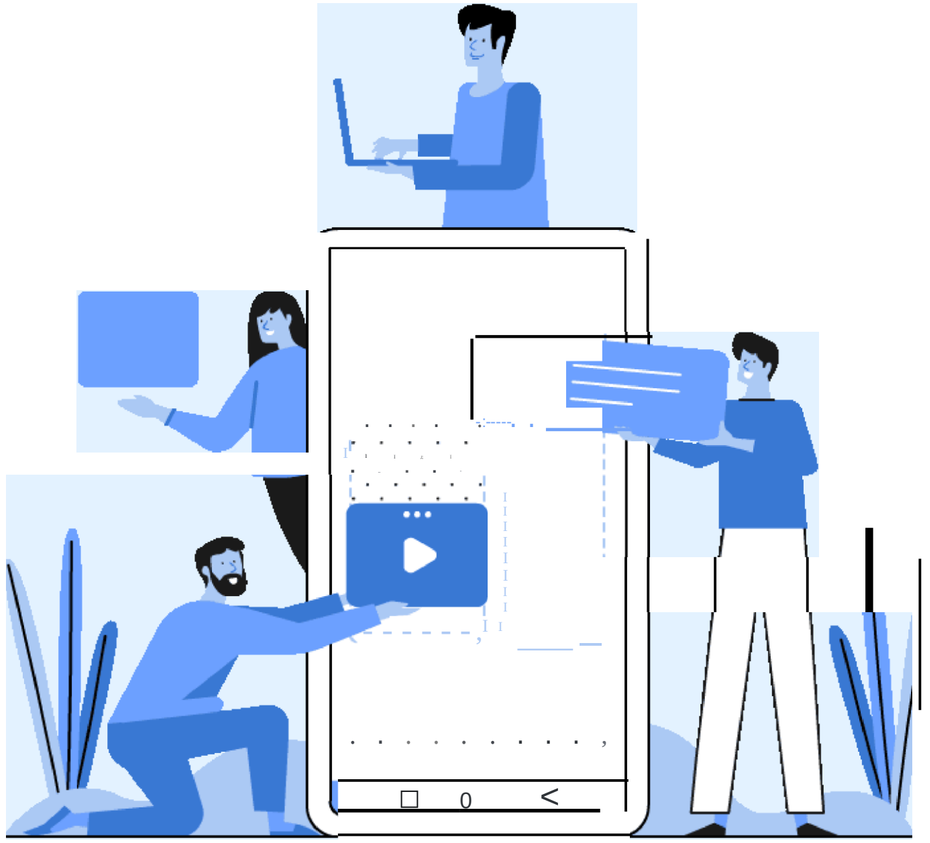
This covers the analysis, achieved objectives and Novelity



Analysis

It provides promotional product results based on non -frequent items in the transaction database that clients are less likely to purchase and maximizes profit by selling the non-frequent items.

The proposed methodology includes operational research theories that only address linear equations, linear programming to begin the profit maximization process, and developing promotional products using the Apriori algorithm to extract common items.



Achieved Ob·ectives

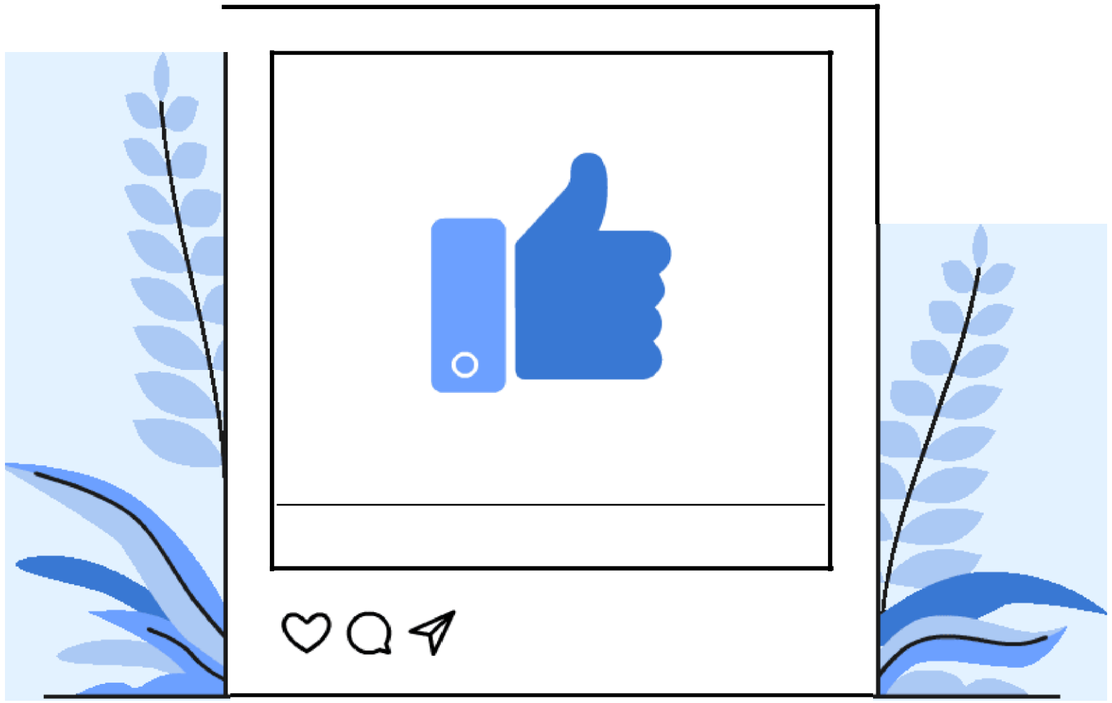
Although Market Basket Analysis conjures up pictures of shopping carts and supermarket shoppers, it is important to realize that there are many other areas in which it can be applied. These include:

- Analysis of credit card purchases
- Analysis of telephone calling patterns
- Identification of fraudulent medical insurance claims
- Providing Better Product Analytics
- Monitoring Market Research
- Analysis of telecom service purchases

We hope to be able to create such recommendation system of patterns that can be used by various retail businesses

Novelity

The Apriori algorithm is more accurate than available algorithms like AIS and SETM algorithm as because both AIS and SETM supports unnecessarily generation and counting of too many candidates item sets that turn out to be small and for each candidate itemset, there are as many entries as its support value which is overcome by the Apriori algorithm.



Thank you

We are happy to take any feedback !!!