GRT INSTITUTE OF ENGINEERING AND TECHNOLOGY, TIRUTTANI – 631209

NATIONAL BOARD OF ACCREDITATION

Approved by AICTE, New Delhi. Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Accredited by NBA, New Delhi

COLLEGE CODE - 1103

NAAN MUDHALVAN – IBM PROJECT (AI)

IBM AI 101 ARTIFICIAL INTELLIGENCE

GROUP-1(TEAM 13)

TITLE : MARKET BASKET INSIGHTS

SUBMITTED BY : **VINOTHKUMAR.T**

3rd Year, 5th Semester

NM ID : (au110321106058)

REG NO : (110321106058)

E-MAIL : vinothkumartedu@gmail.com

DECLARATION

I am **VINOTHKUMAR T** hereby declare that the project report entitled creating a chatbot using python is done byme under the guidance of **Mr.BALAJI K** is submitted inpartial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

Date of submission : 01/11/2023

Place: GRT INSTITUTE OF ENGINEERING AND TECHNOLOGY,TIRUTTANI-631209

T.vinothkumar

SIGNATURE OF THE CANDIDATE

ABSTRACT

ABSTRACT:

In the dynamic landscape of retail, understanding consumer behavior is paramount for businesses seeking to optimize sales strategies and enhance customer experiences. This abstract introduces a comprehensive study focused on market basket insights—a data-driven approach to unraveling the intricate web of consumer purchasing patterns.

Our research delves into the core concepts of market basket analysis, utilizing advanced data mining techniques to extract meaningful associations and trends from transactional data. By examining the contents of shoppers' baskets and identifying frequently co-purchased items, we uncover valuable insights that empower retailers to make informed decisions regarding product placement, promotions, and inventory management.

Moreover, this study showcases the practical applications of market basket insights across various industries, from grocery stores to e-commerce platforms. We explore real-world case studies and success stories, highlighting how organizations have leveraged these insights to boost sales, personalize marketing campaigns, and enhance customer satisfaction.

As consumer preferences continue to evolve, the ability to harness market basket insights becomes increasingly essential for businesses to stay competitive and adapt to changing market dynamics. This research serves as a valuable resource for professionals and researchers in the field of retail analytics, offering a glimpse into the power of data-driven strategies in understanding and influencing consumer behavior.

INTRODUCTION:

Market basket insights, often hailed as the secret sauce of modern retail, offer a profound glimpse into consumer preferences and behavior. In a world where data reigns supreme, understanding what products customers purchase together can unlock a treasure trove of opportunities for businesses. From enhancing cross-selling strategies to improving inventory management, market basket insights are the compass guiding retailers toward greater profitability and customer satisfaction. This introduction lays the found.

Market basket insights, a cornerstone of modern retail analytics, unveil intricate consumer shopping patterns. In essence, it's the art of deciphering what items shoppers tend to purchase together during a single shopping trip. This seemingly mundane data can be a goldmine of strategic information for businesses. By understanding these associations, companies can optimize their operations, boost sales, and enhance the overall customer experience.

In today's hyper-competitive market, where consumers have a plethora of choices, market basket insights are invaluable. They empower retailers to fine-tune their marketing strategies, optimize product placements, and tailor promotions. Moreover, these insights are a potent tool for inventory management, helping businesses minimize wastage and ensure products are in stock when customers want them.

This introduction sets the stage for a deeper exploration of the multifaceted world of market basket insights. From uncovering hidden correlations to shaping personalized recommendations, we'll journey through the vast landscape of data-driven retail optimization. Join us as we delve into the fascinating realm of market basket insights and discover how they're reshaping the way businesses understand and cater to consumer needs.

Problem Definition:

Market basket insights aim to uncover associations and patterns in customer transaction data to understand the co-occurrence of items in a shopping cart. Specifically, it involves:

Data Collection:

Gathering transaction data, typically from point-of-sale systems or e-commerce platforms, including information about items purchased in each transaction.

Association Rule Mining:

Applying data mining techniques, such as Apriori or FP-Growth, to discover associations or rules that indicate which items tend to be bought together. These rules often consist of antecedents (items in the basket) and consequents (items that also appear).

Support, Confidence, and Lift:

Evaluating the strength of these associations using metrics like support (the frequency of occurrence), confidence (the likelihood of the consequent item being purchased given the antecedent), and lift (how much the purchase of the antecedent item affects the purchase of the consequent item).

Insights Generation:

Drawing actionable insights from the discovered associations. For example, identifying which items are frequently purchased together can inform marketing strategies, stock placement, or product bundling.

Business Applications:

Applying these insights to improve business operations, such as optimizing product placement in physical stores, creating targeted marketing campaigns, or suggesting related products online to increase sales and customer satisfaction.

OBJECTIVES:

Increase Cross-Selling:

Identify product combinations frequently purchased together to optimize cross-selling and increase average transaction value.

Inventory Management:

Improve stock management by recognizing which items are commonly bought together, reducing overstock or out-of-stock situations.

Promotion Optimization:

Determine which products should be promoted together to maximize the impact of marketing campaigns and discounts.

Customer Segmentation:
Segment customers based on their purchasing patterns to target them with personalized offers and recommendations.

Store Layout and Design:
Use insights to optimize store layout, placing related items closer to each other to encourage additional purchases.

Supply Chain Efficiency:
Streamline the supply chain by aligning production and distribution with popular product combinations.

Price Strategy:
Adjust pricing strategies based on how items are bundled together in customer transactions.

Loss Prevention:
Detect and prevent fraud or theft by identifying unusual purchasing patterns.

Seasonal Trends:
Understand how customer preferences change with seasons or holidays to plan for inventory and promotions accordingly.

Customer Experience:
Enhance the overall shopping experience by tailoring product recommendations and product placements.

Data Collection:
Gather transactional data from your sales records, including customer IDs, product IDs, and purchase timestamps.
Store the data securely in a database or data warehouse.

Data Preprocessing:
Clean and preprocess the data to handle missing values, outliers, and inconsistencies.Transform the data into a format suitable for analysis.

Market Basket Analysis:
Use association rule mining techniques like or FP-Growth to identify patterns and associations between products frequently bought together.
Calculate metrics like support, confidence, and lift to quantify the significance of these associations.

Customer Segmentation:
Cluster customers based on their purchase behavior or demographics.
Identify segments with distinct shopping patterns.

Visualizations and Reports:

Create visualizations such as scatter plots, and bar charts to present the insights.Generate regular reports or dashboard

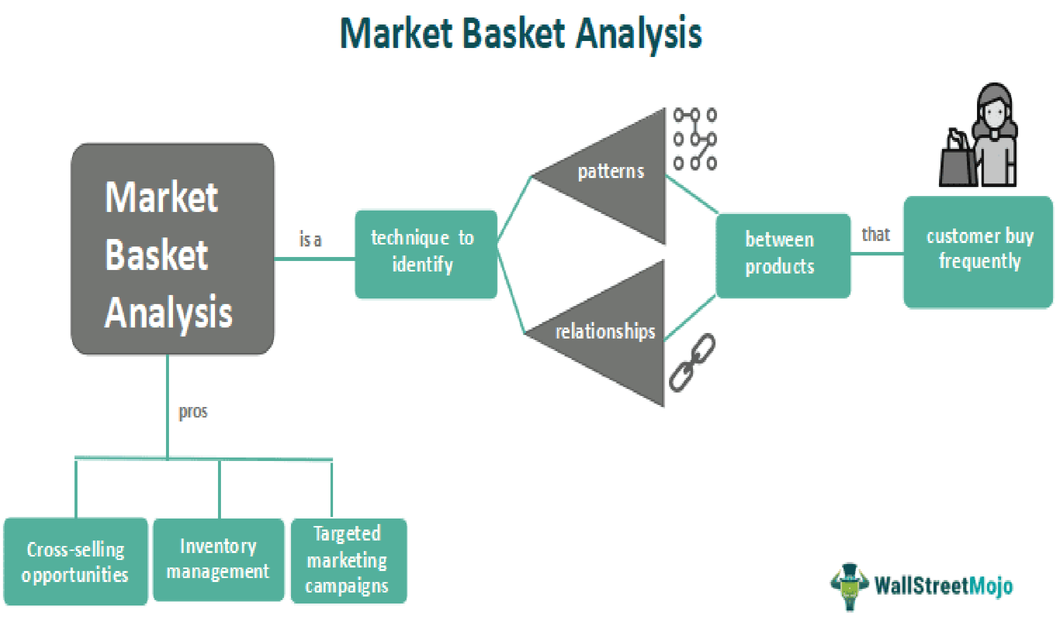
Reduce Churn:

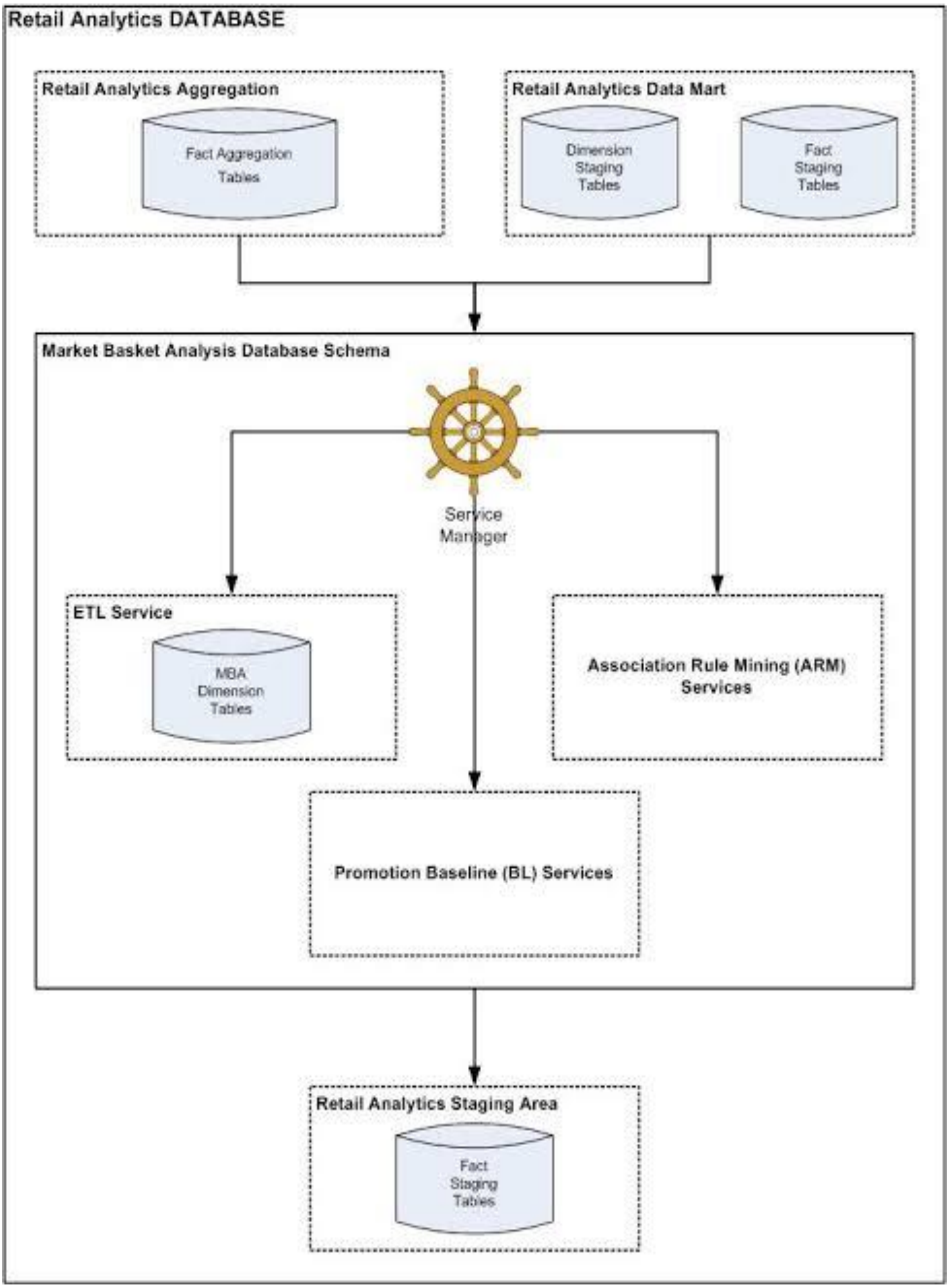
Prevent customer attrition by offering personalized incentives.

Increase Revenue:

Maximize revenue by capitalizing on purchasing patterns.

DESIGN :





INNOVATION :

INNOVATIVE DESIGN ABOUT MARKET BASKET INSIGHTS :

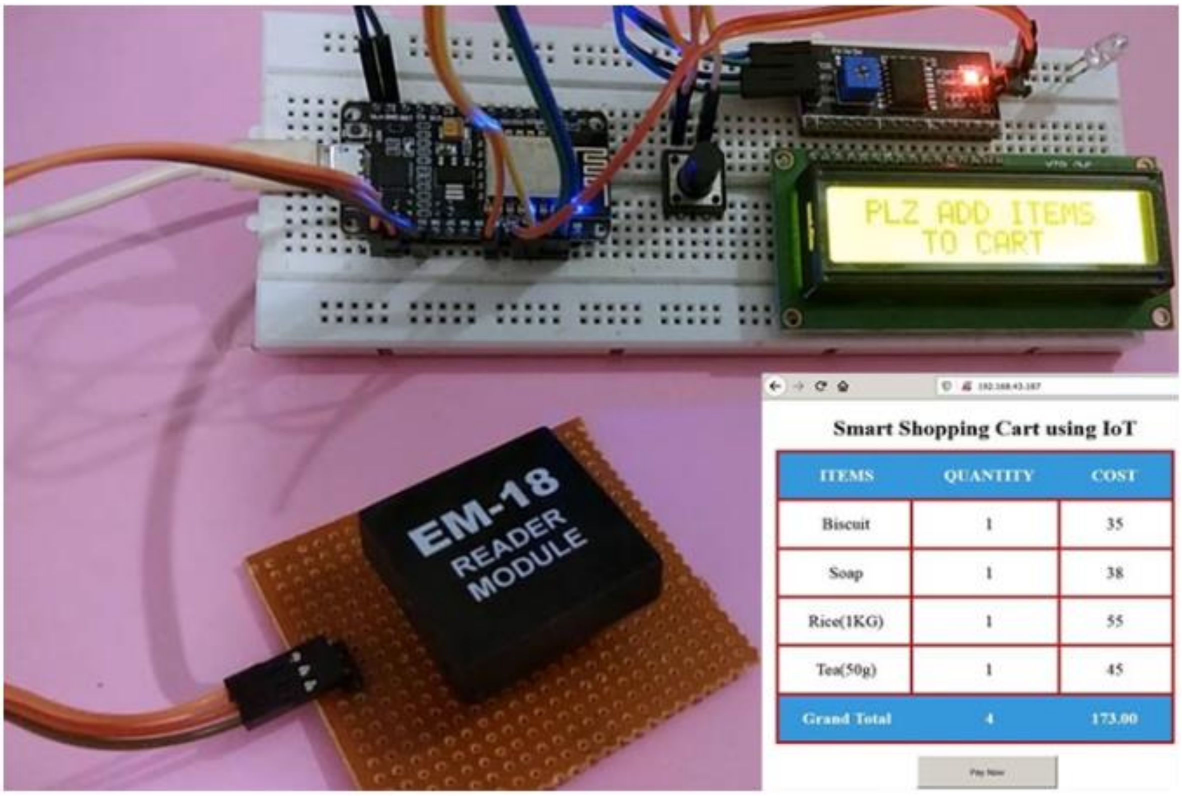
Designing innovations for market basket insights involves creating new methods, tools, or approaches to better understand consumer behaviour, optimize pricing strategies, and enhance the overall shopping experience. Here are some innovative design ideas for market basket insights:

Smart Shopping Carts:

Develop smart shopping carts equipped with RFID technology, cameras, and sensors to automatically scan items as they are placed in the cart. This data can provide real-time insights into customer preferences and item popularity.

Integrate these smart carts with a mobile app that allows customers to view their cart contents and receive personalized recommendations based on their shopping history.

IoT based Smart Shopping Cart using RFID and NodeMCU



AI-Powered Recommendations:

Implement machine learning algorithms to analyze historical shopping data and provide real-time product recommendations to customers as they shop.

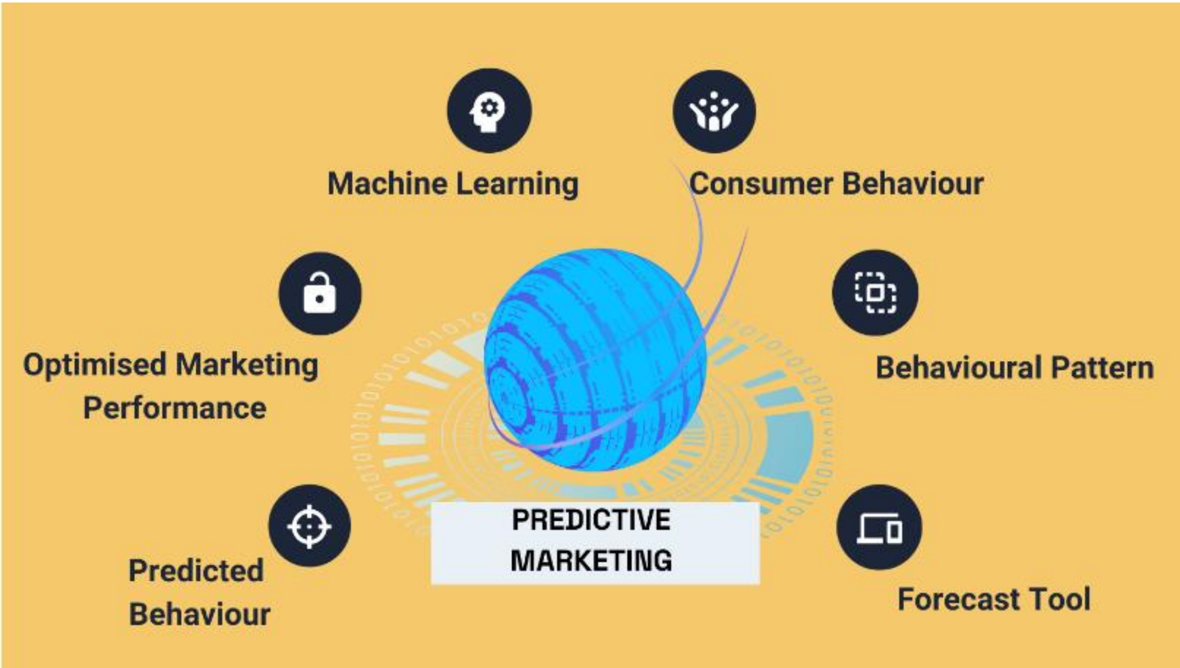
Use natural language processing (NLP) to analyze customer reviews and feedback to enhance product recommendations and improve product descriptions.

Predictive Analytics:

Develop predictive analytics models that forecast future market basket trends and customer behaviours. Retailers can use these insights to optimize inventory management, stock levels, and product placement.

Combine historical sales data with external factors like weather, holidays, and economic indicators to make more accurate predictions.

Predictive Analytics in Marketing:

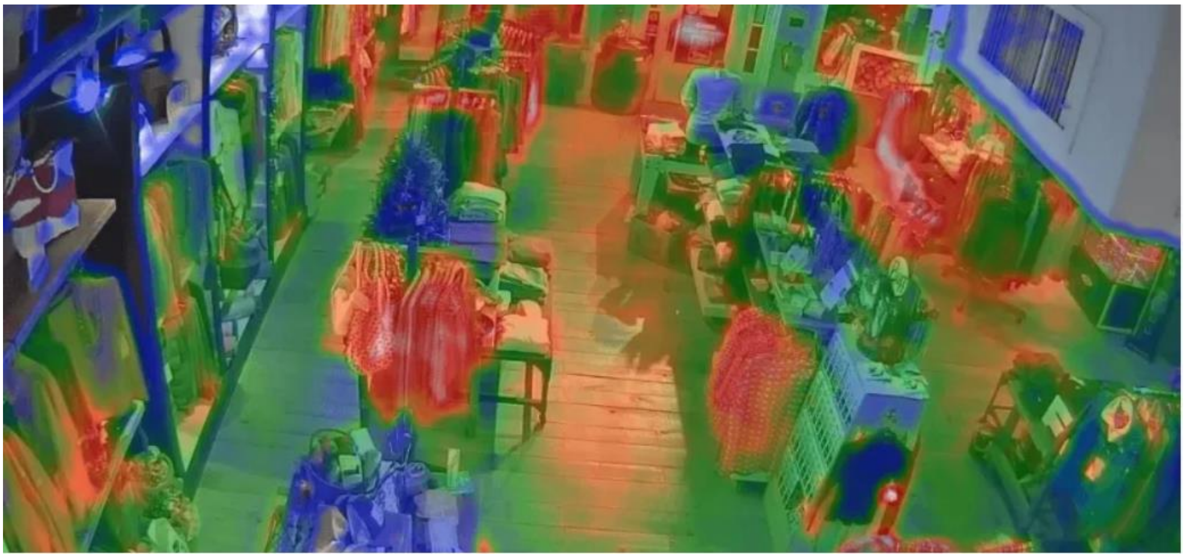


In-Store Heatmaps:

Install cameras and sensors throughout the store to create heatmaps of customer traffic and product interaction. This can help retailers optimize store layouts and product placement.

Analyze these heatmaps to identify popular paths, product touchpoints, and potential areas for improvement.

Heat Map Analysis:



Personalized Pricing Strategies:

Implement dynamic pricing strategies that adjust prices in real-time based on factors such as demand, inventory levels, and customer behavior.

Use machine learning to segment customers and offer personalized discounts and promotions to drive sales.

Dynamic Pricing:

Augmented Reality Shopping:

Create an augmented reality (AR) shopping experience that allows customers to use their smartphones or AR glasses to access additional information about products.

Include features like virtual try-ons, detailed product information, and real-time price comparisons to enhance the shopping experience and encourage cross-selling.

Augmented Reality is the Future of Online Shopping:

Customer Behaviour Analytics:

Develop advanced customer behaviour analytics tools that can analyze data from various touchpoints, including online and offline channels.

Use these tools to identify customer personas, track their journey across channels, and tailor marketing strategies accordingly.

Tracking customer behavior and preferences with data analytics:

Voice and Chatbot Shopping Assistants:

Implement voice-activated shopping assistants or chatbots within the physical store to answer customer queries, provide product information, and make personalized recommendations.

These AI-driven assistants can also help streamline the checkout process.

Data Visualization Dashboards:

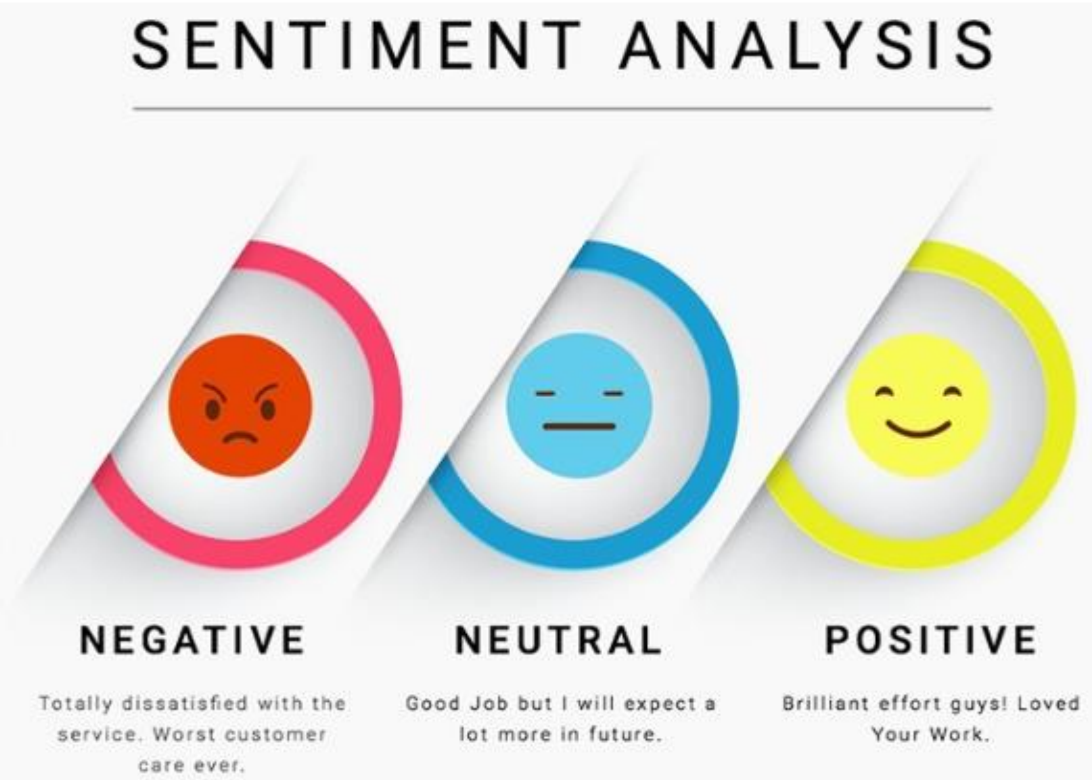
Create user-friendly data visualization dashboards that provide real-time market basket insights to store managers and decision-makers.

Include interactive charts, graphs, and KPIs to help them make data-driven decisions.

Customer Feedback Integration:

Use sentiment analysis to understand customer satisfaction and make immediate improvements based on feedback.

Develop a system for collecting and analyzing customer feedback in real-time. This can be done through surveys, social media monitoring, or in-app feedback forms.



Continuous Monitoring:

Real-time Analytics: Implement real-time monitoring to adapt to changing trends promptly.

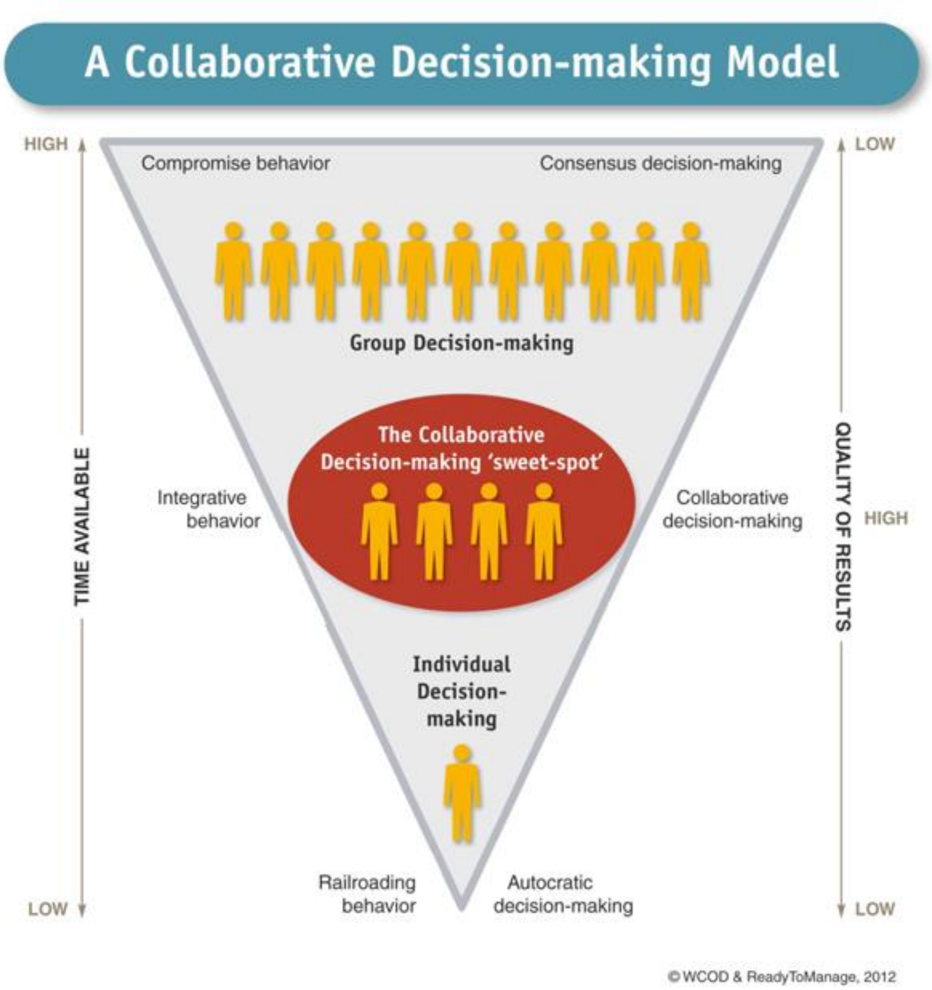
Regular Updates: Periodically update your model and recommendations to reflect evolving customer preferences.

Multivariate Analysis: Incorporate additional factors like pricing, location, and demographics into your analysis to provide more comprehensive insights.

Action Ability:

Collaborative Approach: Involve domain experts, marketing teams, and decision-makers to translate insights into actionable strategies.

Pilot Testing: Test recommendations on a small scale before full-scale implementation.



Development part 1

Introduction :

There are many data analysis tools available to the python analyst and it can be challenging to know which ones to use in a particular situation. A useful (but somewhat overlooked) technique is called association analysis which attempts to find common patterns of items in large data sets. One specific application is often called market basket analysis. The most commonly cited example of market basket analysis is the so-called “beer and diapers” case. The basic story is that a large retailer was able to mine their transaction data and find an unexpected purchase pattern of individuals that were buying beer and baby diapers at the same time.

Unfortunately, this story is most likely a data urban legend. However, it is an illustrative (and entertaining) example of the types of insights that can be gained by mining transactional data.

While these types of associations are normally used for looking at sales transactions; the basic analysis can be applied to other situations like click stream tracking, spare parts ordering and online recommendation engines - just to name a few.

If you have some basic understanding of the python data science world, your first inclination would be to look at scikit-learn for a ready-made algorithm. However, scikit-learn does not support this algorithm. Fortunately, the very useful MLxtend library by Sebastian Raschka has a a an implementation of the Apriori algorithm for extracting frequent item sets for further analysis.

The rest of this article will walk through an example of using this library to analyze a relatively large online retail data set and try to find interesting purchase combinations. By the end of this article, you should be familiar enough with the basic approach to apply it to your own data sets.

Why Association Analysis?

In today’s world, there are many complex ways to analyze data (clustering, regression, Neural Networks, Random Forests, SVM, etc.).The challenge with many of these approaches they can be difficult to tune, challenging to interpret and require quite a bit of data prep and feature engineering to get good results. In other words, they can be very powerful but require a lot of knowledge to implement properly.

Association analysis is relatively light on the math concepts and easy to explain to non-technical people. In addition, it is an unsupervised learning tool that looks for hidden patterns so there is limited need for data prep and feature engineering. It is a good start for certain cases of data exploration and can point the way for a deeper dive into the data using other approaches.

As an added bonus, the python implementation in MLxtend should be very familiar to anyone that has exposure to scikit-learn and pandas. For all these reasons, I think it is a useful tool to be familiar with and can help you with your data analysis problems.

One quick note - technically, market basket analysis is just one application of association analysis. In this post though, I will use association analysis and market basket analysis interchangeably.

Association Analysis 101 :

There are a couple of terms used in association analysis that are important to understand. This chapter in Introduction to Data Mining is a great reference for those interested in the math behind these definitions and the details of the algorithm implementation.

Association rules are normally written like this: {Diapers} -> {Beer} which means that there is a strong relationship between customers that purchased diapers and also purchased beer in the same transaction.

In the above example, the {Diaper} is the **antecedent** and the {Beer} is the **consequent**. Both antecedents and consequents can have multiple items. In other words, {Diaper, Gum} -> {Beer, Chips} is a valid rule.

Support is the relative frequency that the rules show up. In many instances, you may want to look for high support in order to make it is a useful relationship. However, there may be instances where a low support is useful if you are trying to find "hidden" relationships.

Confidence is a measure of the reliability of the rule. A confidence of .5 in the above example would mean that in 50% of the cases where Diaper and Gum were purchased, the purchase also included Beer and Chips. For product recommendation, a 50% confidence may be perfectly acceptable but in a medical situation, this level may not be high enough.

Lift is the ratio of the observed support to that expected if the two rules were independent (see wikipedia). The basic rule of thumb is that a lift value close to 1 means the rules were completely independent. Lift values > 1 are generally more "interesting" and could be indicative of a useful rule pattern.

One final note, related to the data. This analysis requires that all the data for a transaction be included in 1 row and the items should be 1-hot encoded. The MLxtend documentation example is useful:

	Apple	Corn	Dill	Eggs	Ice cream	Kidney Beans	Milk	Nutmeg	Onion	Unicorn	Yogurt
0	0	0	0	1	0	1	1	1	1	0	1
1	0	0	1	1	0	1	0	1	1	0	1
2	1	0	0	1	0	1	1	0	0	0	0
3	0	1	0	0	0	1	1	0	0	1	1
4	0	1	0	1	1	1	0	0	1	0	0

The specific data for this article comes from the UCI Machine Learning Repository and represents transactional data from a UK retailer from 2010-2011. This mostly represents sales to wholesalers so it is slightly different from consumer purchase patterns but is still a useful case study.

Let's Code :

MLxtend can be installed using pip, so make sure that is done before trying to execute any of the code below. Once it is installed, the code below shows how to get it up and running. I have made the notebook available so feel free to follow along with the examples below.

Get our pandas and MLxtend code imported and read the data:

```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules


df = pd.read_excel('http://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online%20Retail.xlsx')
df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

There is a little cleanup, we need to do. First, some of the descriptions have spaces that need to be removed. We'll also drop the rows that don't have invoice numbers and remove the credit transactions (those with invoice numbers containing C).

```
df['Description'] = df['Description'].str.strip()
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
df = df[~df['InvoiceNo'].str.contains('C')]
```

After the cleanup, we need to consolidate the items into 1 transaction per row with each product 1 hot encoded. For the sake of keeping the data set small, I'm only looking at sales for France. However, in additional code below, I will compare these results to sales from Germany. Further country comparisons would be interesting to investigate.

```
basket = (df[df['Country'] =="France"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
```

Here's what the first few columns look like (note, I added some numbers to the columns to illustrate the concept - the actual data in this example is all 0's):

Description	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 EGG HOUSE PAINTED WOOD	12 MESSAGE CARDS WITH ENVELOPES	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS SMALL TUBE RED RETROSPOT	12 PENCILS SMALL TUBE SKULL	12 PENCILS TALL TUBE POSY
InvoiceNo								
536370	11.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
536852	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0
536974	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
537065	0.0	0.0	0.0	0.0	0.0	7.0	0.0	0.0
537463	0.0	0.0	9.0	0.0	0.0	0.0	0.0	0.0

There are a lot of zeros in the data but we also need to make sure any positive values are converted to a 1 and anything less the 0 is set to 0. This step will complete the one hot encoding of the data and remove the postage column (since that charge is not one we wish to explore):

```
def encode_units(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1

basket_sets = basket.applymap(encode_units)
basket_sets.drop('POSTAGE', inplace=True, axis=1)
```

Now that the data is structured properly, we can generate frequent item sets that have a support of at least 7% (this number was chosen so that I could get enough useful examples):

```
frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)
```

The final step is to generate the rules with their corresponding support, confidence and lift:

```
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules.head()
```

	antecedants	consequents	support	confidence	lift
0	(PLASTERS IN TIN WOODLAND ANIMALS)	(PLASTERS IN TIN CIRCUS PARADE)	0.170918	0.597015	3.545907
1	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.168367	0.606061	3.545907
2	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN SPACEBOY)	0.168367	0.530303	3.849607
3	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN CIRCUS PARADE)	0.137755	0.648148	3.849607
4	(PLASTERS IN TIN WOODLAND ANIMALS)	(PLASTERS IN TIN SPACEBOY)	0.170918	0.611940	4.442233

That’s all there is to it! Build the frequent items using `apriori` then build the rules with `association_rules` .

Now, the tricky part is figuring out what this tells us. For instance, we can see that there are quite a few rules with a high lift value which means that it occurs more frequently than would be expected given the number of transaction and product combinations. We can also see several where the confidence is high as well. This part of the analysis is where the domain knowledge will come in handy. Since I do not have that, I'll just look for a couple of illustrative examples.

We can filter the dataframe using standard pandas code. In this case, look for a large lift (6) and high confidence (.8):

```
rules[ (rules['lift'] >= 6) &
       (rules['confidence'] >= 0.8) ]
```

	antecedants	consequents	support	confidence	lift
8	(SET/6 RED SPOTTY PAPER CUPS)	(SET/6 RED SPOTTY PAPER PLATES)	0.137755	0.888889	6.968889
9	(SET/6 RED SPOTTY PAPER PLATES)	(SET/6 RED SPOTTY PAPER CUPS)	0.127551	0.960000	6.968889
10	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.096939	0.815789	8.642959
11	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.094388	0.837838	8.642959
16	(SET/6 RED SPOTTY PAPER CUPS, SET/6 RED SPOTTY...	(SET/20 RED RETROSPOT PAPER NAPKINS)	0.122449	0.812500	6.125000
17	(SET/6 RED SPOTTY PAPER CUPS, SET/20 RED RETRO...	(SET/6 RED SPOTTY PAPER PLATES)	0.102041	0.975000	7.644000
18	(SET/6 RED SPOTTY PAPER PLATES, SET/20 RED RET...	(SET/6 RED SPOTTY PAPER CUPS)	0.102041	0.975000	7.077778
22	(SET/6 RED SPOTTY PAPER PLATES)	(SET/20 RED RETROSPOT PAPER NAPKINS)	0.127551	0.800000	6.030769

In looking at the rules, it seems that the green and red alarm clocks are purchased together and the red paper cups, napkins and plates are purchased together in a manner that is higher than the overall probability would suggest.

At this point, you may want to look at how much opportunity there is to use the popularity of one product to drive sales of another. For instance, we can see that we sell 340 Green Alarm clocks but only 316 Red Alarm Clocks so maybe we can drive more Red Alarm Clock sales through recommendations?

```
basket['ALARM CLOCK BAKELIKE GREEN'].sum()

340.0

basket['ALARM CLOCK BAKELIKE RED'].sum()

316.0
```

What is also interesting is to see how the combinations vary by country of purchase. Let's check out what some popular combinations might be in Germany:

```
basket2 = (df[df['Country'] =="Germany"]
```

```
        .groupby(['InvoiceNo', 'Description'])['Quantity']
        .sum().unstack().reset_index().fillna(0)
        .set_index('InvoiceNo'))

basket_sets2 = basket2.applymap(encode_units)
basket_sets2.drop('POSTAGE', inplace=True, axis=1)
frequent_itemsets2 = apriori(basket_sets2, min_support=0.05, use_colnames=True)
rules2 = association_rules(frequent_itemsets2, metric="lift", min_threshold=1)

rules2[ (rules2['lift'] >= 4) &
       (rules2['confidence'] >= 0.5)]
```

	antecedants	consequents	support	confidence	lift
7	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.107221	0.571429	4.145125
9	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.115974	0.584906	4.242887
10	(RED RETROSPOT CHARLOTTE BAG)	(WOODLAND CHARLOTTE BAG)	0.070022	0.843750	6.648168

It seems that in addition to David Hasselhoff, Germans love Plasters in Tin Spaceboy and Woodland Animals.

In all seriousness, an analyst that has familiarity with the data would probably have a dozen different questions that this type of analysis could drive. I did not replicate this analysis for additional countries or customer combos but the overall process would be relatively simple given the basic pandas code shown above.

ITEM CO-OCCURRENCE ANALYSIS :

Explore which items tend to co-occur in the same transaction.
This can help with store layout optimization and product placement.
Develop strategies to cross-sell items based on the association rules. For example, if a customer adds bread to their cart, suggest butter or jam.

CUSTOMER SEGMENTATION :

Segment customers based on their purchasing behavior and analyze the frequent itemsets and association rules for each segment.To perform these analyses, you'll need to adapt your code and data manipulation based on the specific questions and insights you're looking to extract from the dataset. Additionally, you can utilize data visualization tools and techniques to present your findings effectively.

Development part 2

Topic :

In this technology you will continue building your project by selecting a machine leairning algorithm, training the model , and evaluating its performance.

Perform different analysis as needed. After performing the relevant activities create a document around it and share the same for assessment.

Data Source :

A good data source for market basket analysis using analysis techniques ,Apriori algorithm to find frequently co-occuring products and generate insights for business optimization.

Dataset Link :

(https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis)

BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
536365	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850	United Kingdom
536365	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850	United Kingdom
536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	SET 7 BABUSHKA NESTING BOXES	2	12/1/2010 8:26	7.65	17850	United Kingdom
536365	GLASS STAR FROSTED T-LIGHT HOLDER	6	12/1/2010 8:26	4.25	17850	United Kingdom
536366	HAND WARMER UNION JACK	6	12/1/2010 8:28	1.85	17850	United Kingdom
536366	HAND WARMER RED POLKA DOT	6	12/1/2010 8:28	1.85	17850	United Kingdom
536367	ASSORTED COLOUR BIRD ORNAMENT	32	12/1/2010 8:34	1.69	13047	United Kingdom
536367	POPPY'S PLAYHOUSE BEDROOM	6	12/1/2010 8:34	2.1	13047	United Kingdom
536367	POPPY'S PLAYHOUSE KITCHEN	6	12/1/2010 8:34	2.1	13047	United Kingdom
536367	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	12/1/2010 8:34	3.75	13047	United Kingdom
536367	IVORY KNITTED MUG COSY	6	12/1/2010 8:34	1.65	13047	United Kingdom
536367	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	12/1/2010 8:34	4.25	13047	United Kingdom
536367	BOX OF VINTAGE JIGSAW BLOCKS	3	12/1/2010 8:34	4.95	13047	United Kingdom
536367	BOX OF VINTAGE ALPHABET BLOCKS	2	12/1/2010 8:34	9.95	13047	United Kingdom
536367	HOME BUILDING BLOCK WORD	3	12/1/2010 8:34	5.95	13047	United Kingdom
536367	LOVE BUILDING BLOCK WORD	3	12/1/2010 8:34	5.95	13047	United Kingdom
536367	RECIPE BOX WITH METAL HEART	4	12/1/2010 8:34	7.95	13047	United Kingdom
536367	DOORMAT NEW ENGLAND	4	12/1/2010 8:34	7.95	13047	United Kingdom
536368	JAM MAKING SET WITH JARS	6	12/1/2010 8:34	4.25	13047	United Kingdom
536368	RED COAT RACK PARIS FASHION	3	12/1/2010 8:34	4.95	13047	United Kingdom
536368	YELLOW COAT RACK PARIS FASHION	3	12/1/2010 8:34	4.95	13047	United Kingdom
536368	BLUE COAT RACK PARIS FASHION	3	12/1/2010 8:34	4.95	13047	United Kingdom
536369	BATH BUILDING BLOCK WORD	3	12/1/2010 8:35	5.95	13047	United Kingdom
536370	ALARM CLOCK BAKELIKE PINK	24	12/1/2010 8:45	3.75	12583	France
536370	ALARM CLOCK BAKELIKE RED	24	12/1/2010 8:45	3.75	12583	France
536370	ALARM CLOCK BAKELIKE GREEN	12	12/1/2010 8:45	3.75	12583	France

retaildata

Machine learning algorithms :

Market basket analysis is a common application of machine learning in retail and e-commerce to discover patterns and associations between items that are frequently purchased together.

The most popular algorithm for market basket analysis is the Apriori algorithm. However, there are other techniques and variations that can be used, depending on the specific requirements and size of your dataset. Here are some popular choices:

Apriori Algorithm :

Apriori is a classic algorithm for association rule mining, particularly for market basket analysis. It identifies frequent itemsets and generates association rules based on support and confidence levels.

FP-growth Algorithm :

The FP-growth (Frequent Pattern growth) algorithm is an alternative to Apriori that is more efficient in terms of memory and runtime. It builds a compact data structure called an FP-tree to mine frequent itemsets.

Eclat Algorithm :

Eclat (Equivalence Class Transformation) is another algorithm for frequent itemset mining. It uses a depth- first search approach and is known for its simplicity and efficiency.

FPGrowth Algorithm :

FPGrowth (Frequent Pattern Growth) is a variation of FP-growth that works well with large datasets and is implemented in libraries like Spark's MLlib.

Training the model :

Training a machine learning model, regardless of the specific algorithm you choose, involves several key steps. Here is a high-level overview of the typical process for training a machine learning model.

Data Collection :

Gather and prepare a dataset that includes historical or training data. This data should consist of input features (attributes) and corresponding output labels or target values that the model needs to learn to predict.

Data Preprocessing :

Clean and preprocess the data to ensure it is in a suitable format for training. This may involve tasks such as handling missing values, encoding categorical variables, scaling features, and splitting the data into training and testing sets.

Feature Engineering :

Depending on the specific problem and dataset, you may need to engineer or create new features that can improve the model's ability to learn patterns and make predictions effectively.

Choosing a Model :

Select the machine learning algorithm or model architecture that is appropriate for your problem. The choice of model depends on factors like the type of data, the nature of the problem (classification, regression, clustering, etc.), and your specific goals.

Model Training :

Train the selected model using the training data. During training, the model learns to make predictions by adjusting its internal parameters to minimize a predefined loss or error function. This involves iterations or epochs, and the model gradually improves its performance.

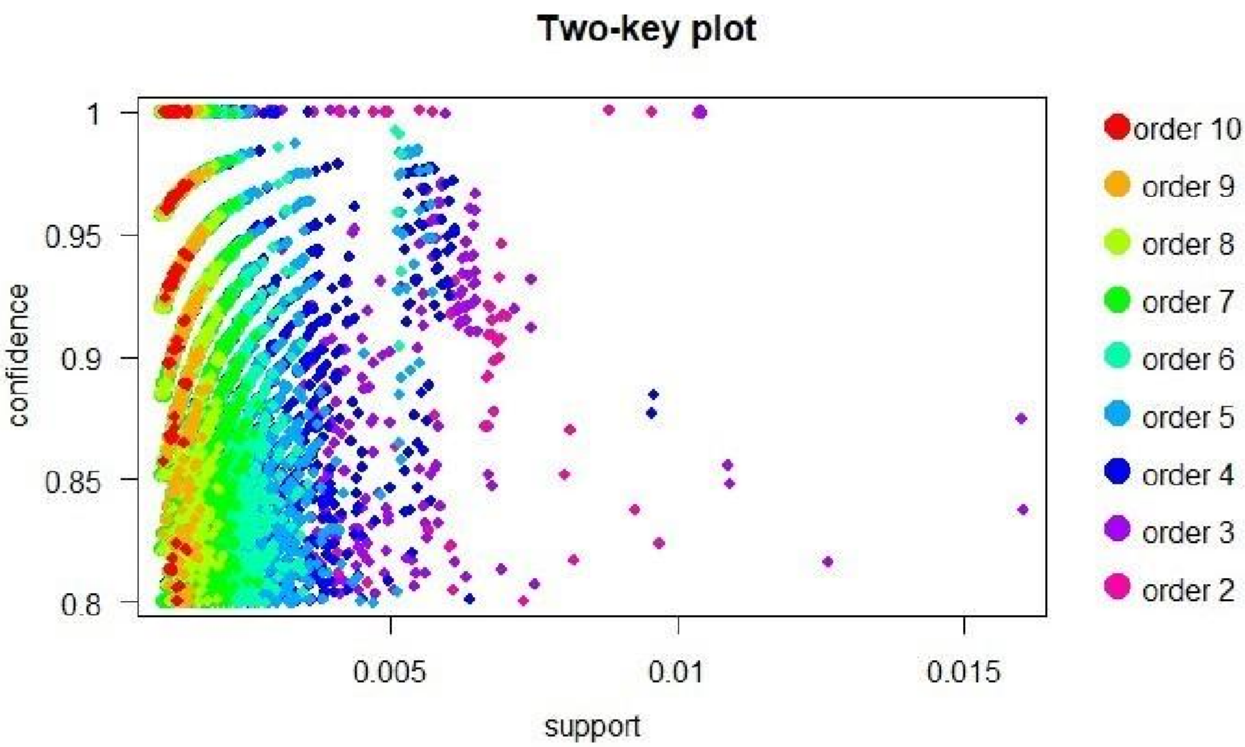
Python program :

```
  import pandas as pd
  from mlxtend.frequent_patterns import apriori
  from mlxtend.frequent_patterns import association_rules

# Load your dataset
data = pd.read_csv('your_dataset.csv')

# Data Preprocessing
# You may need to preprocess your dataset to create a binary matrix#where
columns represent items, and rows represent transactions.

# perform Association Analysis

# Use Apriori to find frequent itemsets
```

```
frequent_itemsets = apriori(data, min_support=0.1, use_colnames=True)

# Generate Association Rules

association_rules = association_rules(frequent_itemsets, metric="lift",min_threshold=1.0)

# Display the association rules

print(association_rules)
```

OUTPUT :

	antecedents	consequents	antecedent support	...	lift
0	(Crispy Potato Chips)	(Refreshing Cola)	0.3	...	1.2
1	(Refreshing Cola)	(Crispy Potato Chips)	0.4	...	1.2
2	(Chocolate Ice Cream)	(Freshly Baked Baguette)	0.2	...	0.8
3	(Freshly Baked Baguette)	(Chocolate Ice Cream)	0.3	...	0.8
4	(Crispy Potato Chips, Refreshing Cola)	(Chocolate Ice Cream)	0.1	...	1.5
5	(Crispy Potato Chips, Chocolate Ice Cream)	(Refreshing Cola)	0.1	...	1.25
6	(Refreshing Cola, Chocolate Ice Cream)	(Crispy Potato Chips)	0.1	...	1.5
7	(Crispy Potato Chips)	(Refreshing Cola, Chocolate Ice Cream)	0.3	...	1.5



Evaluate the performance of the algorithm :

The performance of the Apriori algorithm in market basket analysis can vary based on several factors, including the size of the dataset, the hardware and software used, and the specific parameters and implementation of the algorithm. Here are some considerations regarding the performance of the Apriori algorithm in market basket insights.

Scalability :

Apriori can be computationally expensive, especially when dealing with large transaction datasets with many items. The algorithm has to generate a large number of candidate itemsets, and this process can become slow as the dataset size increases.

Thresholds :

The performance of Apriori is influenced by the minimum support and confidence thresholds you set. Lower support thresholds can result in more frequent itemsets but may increase computational complexity. Finding the right balance is essential.

Data Preprocessing :

Data preprocessing, such as reducing the number of unique items or filtering out infrequent items, can significantly impact the algorithm's performance. Cleaning the data and removing noise is important.

Algorithm Optimization :

There are various optimization techniques and variations of the Apriori algorithm that can improve its performance, such as the use of hash-based techniques and pruning strategies. Choosing an optimized implementation can make a significant difference.