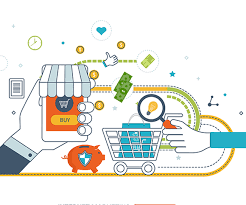
**MARKET BASKET INSIGHTS**



**Project Title: MARKET BASKET INSIGHTS**

Market Basket Insights refers to the process of analyzing transactional data, typically from retail or e-commerce environments, to uncover patterns and associations among the items that customers purchase together. This analysis aims to reveal valuable insights that businesses can use to improve their operations and strategies. Key components of Market Basket Insights include:

1.Association Analysis: This involves identifying relationships and associations between products or items frequently purchased together. It helps businesses understand the co-occurrence patterns of items in customers' baskets.

2.Recommendation Systems: Market Basket Insights can be used to create personalized product recommendations for customers based on their historical purchase behavior, increasing cross-selling and customer satisfaction.

3.Inventory Management: By understanding which products are often bought together, businesses can optimize their inventory management, reduce overstock or understock situations, and improve supply chain efficiency.

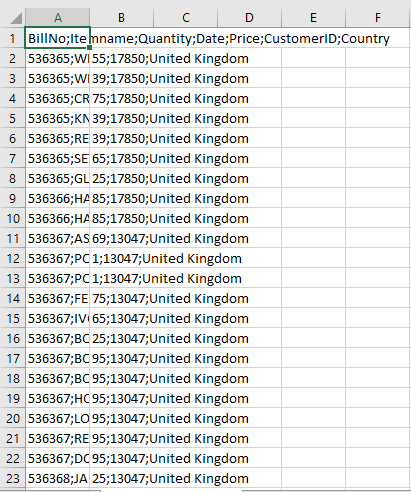
4.Promotion Planning: This analysis can inform businesses about which products are strong candidates for bundling or promotional campaigns, ultimately driving sales and revenue.

5.Customer Segmentation: Market Basket Insights can help categorize customers into segments based on their purchase behavior, allowing for more targeted marketing and personalized offers.

6.Market Basket Analysis: This is the core technique used in Market Basket Insights. It involves the application of data mining and machine learning methods to identify frequent itemsets and association rules that reveal purchasing patterns.

Overall, Market Basket Insights provides businesses with actionable information that can enhance their decision-making, sales strategies, and customer experiences by harnessing the power of transactional data.

**Dataset Link:**[**https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis**](https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis)



To gain market basket insights, you can follow these steps:

1. Collect transactional data. This data should include information about customer purchases, such as the items purchased, the time and date of the transaction, and any other relevant information. You can obtain this data from various sources, such as point-of-sale systems, e-commerce platforms, or customer loyalty programs.
2. Prepare the data. This may involve cleaning the data to remove any errors or inconsistencies, and transforming the data into a format that can be easily analyzed.
3. Identify frequent itemsets. This is the process of identifying sets of items that are frequently purchased together. A variety of data mining algorithms can be used for this task, such as Apriori or FP-Growth.
4. Generate association rules. Association rules are statements about the relationships between different itemsets. For example, an association rule might be "If a customer buys bread, they are also likely to buy milk." Association rules can be generated using the frequent itemsets identified in the previous step.
5. Evaluate the association rules. Once association rules have been generated, they need to be evaluated to determine which ones are meaningful and actionable. This can be done by considering the support and confidence of each rule. Support is the percentage of transactions that contain the itemset, while confidence is the percentage of transactions that contain the antecedent itemset that also contain the consequent itemset.
6. Generate insights and recommendations. Once the association rules have been evaluated, you can use them to generate insights and recommendations for your business. For example, you might use the association rules to identify opportunities for cross-selling and upselling, or to develop new product bundles.

Here are some additional tips for conducting market basket analysis:

* Use a large enough dataset. The larger the dataset, the more accurate your results will be.
* Set appropriate support and confidence thresholds. The support threshold determines how often an itemset must appear in the data to be considered frequent, while the confidence threshold determines how strong the relationship between the antecedent and consequent itemsets must be.
* Consider the context of the data. When interpreting the results of market basket analysis, it is important to consider the context of the data. For example, if you are analyzing data from a grocery store, you might expect to see association rules between food items. However, if you are analyzing data from a clothing store, you might expect to see association rules between different types of clothing items.

Market basket analysis can be a powerful tool for understanding customer behavior and identifying opportunities to improve your business. By following the steps above, you can gain valuable insights from your transactional data.

tuneshare

more\_vert

**Necessary step to follow:**

**1.Import Libraries:**

Start by importing the necessary libraries:

**Program:**

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

**2.Load the Dataset:**

Load your dataset into a Pandas DataFrame. You can typically find

house price datasets in CSV format, but you can adapt this code to other

formats as needed.

**Program:**

df = pd.read\_csv(' E:\USA\_Housing.csv ')

Pd.read()

**3. Exploratory Data Analysis (EDA):**

Perform EDA to understand your data better. This includes

checking for missing values, exploring the data's statistics, and

visualizing it to identify patterns.

**Program:**

# Check for missing values

print(df.isnull().sum())

# Explore statistics

print(df.describe())

# Visualize the data (e.g., histograms, scatter plots, etc.)

**4. Feature Engineering:**

Depending on your dataset, you may need to create new features or

transform existing ones. This can involve one-hot encoding categorical

variables, handling date/time data, or scaling numerical features.

**Program:**

# Example: One-hot encoding for categorical variables

df = pd.get\_dummies(df, columns=[' Avg. Area Income ', ' Avg. Area

House Age '])

**5. Split the Data:**

Split your dataset into training and testing sets. This helps you evaluate

your model's performance later.

X = df.drop('price', axis=1) # Features

y = df['price'] # Target variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,

random\_state=42)

**6. Feature Scaling:**

Apply feature scaling to normalize your data, ensuring that all

features have similar scales. Standardization (scaling to mean=0 and

std=1) is a common choice.

**Program:**

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

**Importance of loading and processing dataset:**

Loading and preprocessing the dataset is an important first step in

building any machine learning model. However, it is especially

important for house price prediction models, as house price datasets are

often complex and noisy.

By loading and preprocessing the dataset, we can ensure that the

machine learning algorithm is able to learn from the data effectively and

accurately.

**Challenges involved in loading and preprocessing a house price dataset:**

**There are a number of challenges involved in loading and preprocessing**

**a house price dataset, including:**

* **Handling missing values:**

House price datasets often contain missing values, which can

be due to a variety of factors, such as human error or incomplete data

collection. Common methods for handling missing values include

dropping the rows with missing values, imputing the missing values with

the mean or median of the feature, or using a more sophisticated method

such as multiple imputation.

* **Encoding categorical variables:**

House price datasets often contain categorical features, such as the

type of house, the neighborhood, and the school district. These features

need to be encoded before they can be used by machine learning models.

One common way to encode categorical variables is to use one-hot

encoding.

* **Scaling the features:**

It is often helpful to scale the features before training a

machine learning model. This can help to improve the performance of

the model and make it more robust to outliers. There are a variety of

ways to scale the features, such as min-max scaling and standard scaling.

* **Splitting the dataset into training and testing sets:**

Once the data has been pre-processed, we need to split the

dataset into training and testing sets. The training set will be used to

train the model, and the testing set will be used to evaluate the

performance of the model on unseen data. It is important to split the

dataset in a way that is representative of the real world distribution of the

data.

**How to overcome the challenges of loading and preprocessing a**

**house price dataset:**

There are a number of things that can be done to overcome the

challenges of loading and preprocessing a house price dataset, including:

* **Use a data preprocessing library:**

There are a number of libraries available that can help with data

preprocessing tasks, such as handling missing values, encoding

categorical variables, and scaling the features.

* **Carefully consider the specific needs of your model:**

The best way to preprocess the data will depend on the specific

machine learning algorithm that you are using. It is important to

carefully consider the requirements of the algorithm and to preprocess

the data in a way that is compatible with the algorithm.

* **Validate the preprocessed data:**

It is important to validate the preprocessed data to ensure that it is

in a format that can be used by the machine learning algorithm and that

it is of high quality. This can be done by inspecting the data visually or

by using statistical methods.

1.Loading the dataset:

* Loading the dataset using machine learning is the process of bringing

the data into the machine learning environment so that it can be used

to train and evaluate a model.

* The specific steps involved in loading the dataset will vary depending

on the machine learning library or framework that is being used.

However, there are some general steps that are common to most

machine learning frameworks:

**a.Identify the dataset:**

The first step is to identify the dataset that you want to load. This

dataset may be stored in a local file, in a database, or in a cloud storage

service.

**b.Load the dataset:**

Once you have identified the dataset, you need to load it into the

machine learning environment. This may involve using a built-in

function in the machine learning library, or it may involve writing your

own code.

**c.Preprocess the dataset:**

Once the dataset is loaded into the machine learning environment,

you may need to preprocess it before you can start training and

evaluating your model. This may involve cleaning the data, transforming

the data into a suitable format, and splitting the data into training and

test sets.

Here, how to load a dataset using machine learning in Python

Program

import pandas as pd

from mlxtend.frequent\_patterns import apriori, association\_rules

pd.set\_option('display.max\_columns', None)

pd.set\_option('display.max\_rows', None)

pd.set\_option('display.width', 500)

df = pd.read\_excel("/kaggle/input/market-basket-analysis/Assignment-1\_Data.xlsx")

def outlier\_thresholds(dataframe, variable):

quartile1 = dataframe[variable].quantile(0.01)

quartile3 = dataframe[variable].quantile(0.99)

interquantile\_range = quartile3 - quartile1

up\_limit = quartile3 + 1.5 \* interquantile\_range

low\_limit = quartile1 - 1.5 \* interquantile\_range

return low\_limit, up\_limit

def replace\_with\_thresholds(dataframe, variable):

low\_limit, up\_limit = outlier\_thresholds(dataframe, variable)

dataframe.loc[(dataframe[variable] < low\_limit), variable] = low\_limit

dataframe.loc[(dataframe[variable] > up\_limit), variable] = up\_limit

def retail\_data\_prep(dataframe):

dataframe = dataframe[dataframe["Quantity"] > 0]

dataframe = dataframe[dataframe["Price"] > 0]

replace\_with\_thresholds(dataframe, "Quantity")

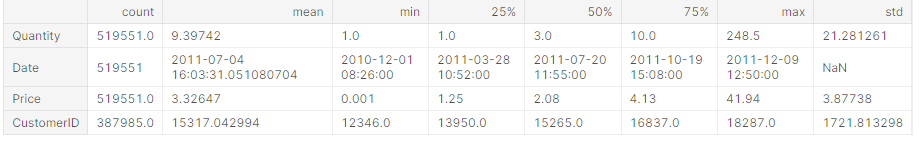
replace\_with\_thresholds(dataframe, "Price")

return dataframe

df = retail\_data\_prep(df)

df.describe().T

output

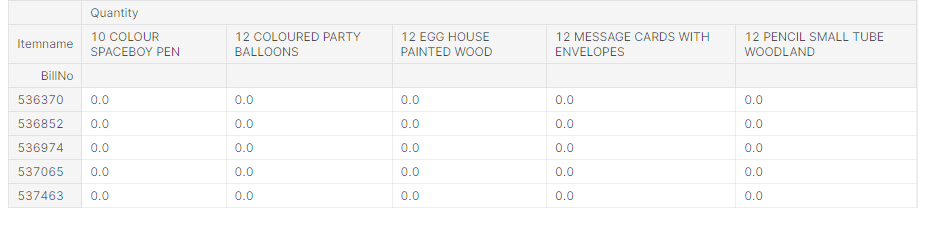


Program

df\_fr = df[df['Country'] == "France"]

df\_fr.groupby(['BillNo', 'Itemname']).agg({"Quantity": "sum"}).unstack().fillna(0).iloc[0:5, 0:5]

output



Program

fr\_inv\_pro\_df=df\_fr.groupby(['BillNo', 'Itemname']). \

agg({"Quantity": "sum"}). \

unstack(). \

fillna(0). \

applymap(lambda x: 1 if x > 0 else 0)frequent\_itemsets = apriori(fr\_inv\_pro\_df.astype("bool"), min\_support=0.01, use\_colnames=True)

frequent\_itemsets.sort\_values("support", ascending=False).head()

Output



Conclusion

Market basket insights provide valuable information for businesses, retailers, and marketers. By analyzing customer purchase patterns and associations between products, businesses can make informed decisions to improve their marketing strategies, inventory management, and overall customer experience. Here are some key takeaways:

Improved Customer Experience: Understanding which products are often purchased together allows businesses to offer personalized recommendations and create a more tailored shopping experience. This can lead to increased customer satisfaction and loyalty.

Optimized Inventory Management: Market basket insights help businesses optimize their inventory by stocking products that are frequently purchased together. This reduces waste and ensures that popular product combinations are readily available.

Targeted Marketing: With a better understanding of customer preferences, businesses can create more effective marketing campaigns. They can target specific customer segments with relevant product recommendations and promotions.

Cross-Selling and Upselling Opportunities: Market basket analysis identifies opportunities for cross-selling and upselling. Businesses can promote complementary products to increase the average transaction value.

Price Optimization: By analyzing price sensitivity and product associations, businesses can fine-tune their pricing strategies, maximizing revenue without alienating customers.

Seasonal and Trend Analysis: Market basket insights can reveal changing customer preferences and seasonal trends, allowing businesses to adapt their offerings and marketing efforts accordingly.

Competitive Advantage: Businesses that effectively leverage market basket insights gain a competitive advantage by staying ahead of customer demands and market trends.

In summary, market basket insights are a valuable tool for data-driven decision-making in retail and other industries. They enable businesses to understand customer behavior, optimize operations, and ultimately enhance the overall shopping experience. By continually analyzing and acting upon these insights, businesses can adapt to evolving market dynamics and build stronger customer relationships.