Market Basket Insights

INTRODUCTION :

In today's competitive retail landscape, understanding customer behavior is paramount to staying ahead of the curve. "Market Basket Insights" is a data-driven initiative designed to unravel the intricate web of customer purchasing behavior through the application of Association Analysis. By leveraging this powerful analytical technique on a carefully curated dataset, our project aims to illuminate hidden patterns and associations among products, ultimately empowering the retail business to make informed decisions, enhance the shopping experience, and drive revenue growth

Objective: The primary objectives of this project are as follows: • Uncover Hidden Patterns: Through Association Analysis, we will identify patterns of product co-occurrence in customer transactions. This will enable us to reveal frequently purchased items that might not be obvious through simple observation

• **Understand Customer Behavior:** By analyzing transaction data, we will gain insights into customer preferences, affinities, and buying habits. This understanding is invaluable for tailoring marketing strategies and optimizing product placement.

• **Identify Cross-Selling Opportunities**: Armed with knowledge about product associations, the project seeks to identify strategic cross-selling opportunities. Recommending complementary products to customers can enhance their shopping experience and increase the average transaction value.

Methodology: Our project's methodology includes the following key steps:

• Data Collection: We will gather transactional data from the retail business, including details about products purchased together in each transaction, along with relevant customer information.

• **Data Preprocessing**: The collected data will undergo preprocessing, including cleaning, transformation, and encoding, to ensure it is suitable for analysis.

• **Association Analysis**: Utilizing algorithms such as Apriori or FP-growth, we will perform Association Analysis to discover product associations and generate association rules. These rules will highlight which products are frequently bought together.

- **Insights Generation:** The discovered patterns and association rules will be interpreted to gain actionable insights into customer behavior and preferences.
- **Cross-Selling Recommendations**: Based on the insights gained, we will develop strategies and recommendations for cross-selling products to customers, potentially increasing revenue and customer satisfaction.
- **Visualization:** Visualization tools and techniques will be employed to present the results in an easily understandable and actionable format.

GIVEN DATASET:

Certainly, to perform market basket analysis, you'll need a dataset of transaction records. Each record should represent a single transaction, and the items purchased in each transaction should be recorded. Here's an example of what a simple transaction dataset might look like:

Transaction ID	Items
1	Bread, Milk, Eggs
2	Bread, Juice
3	Milk, Juice, Diapers
4	Bread, Milk, Juice
5	Eggs, Diapers
6	Bread, Milk, Juice
7	Milk, Eggs
8	Bread, Diapers
9	Juice
10	Bread, Milk, Diapers

In this dataset, "Transaction ID" is a unique identifier for each transaction, and the "Items" column lists the items purchased in each transaction.

You can use this dataset to apply market basket analysis techniques such as association rule mining (e.g., Apriori algorithm) to discover relationships between items, identify frequently co-purchased products, and make data-driven decisions to optimize your sales and marketing strategies.

OVERVIEW OF THE PROCESS:

Market basket insights, often obtained through market basket analysis, is a process used to discover patterns and relationships between products that are frequently purchased together by customers. Here's an overview of the process:

1.DataCollection:

Gather transaction data that records what items customers purchase in each transaction. Each transaction should be associated with a unique identifier.

2. Data Preprocessing:

- Remove duplicates: Ensure that duplicate transactions are removed, especially if customers' shopping habits are tracked more than once.
- Encode data: Convert the data into a suitable format, such as a binary matrix, where each row represents a transaction, and each column represents an item. A "1" can indicate the presence of an item in a transaction, and "0" indicates its absence.

3.Support Calculation: Calculate the support for each itemset. Support is the proportion of transactions that contain a specific itemset. High support indicates the itemset is frequently bought together.

4.Frequent Itemset Generation:

- Start with single items as frequent itemsets.
- Generate new itemsets by joining and pruning based on the support threshold.
- Continue this process to find all frequent itemsets.

5.Association Rule Mining:

- From frequent itemsets, generate association rules. An association rule has an antecedent (the item or items on the left) and a consequent (the item on the right).
- Calculate confidence and lift for each rule. Confidence measures how often the rule is true, and lift indicates the strength of the association.

6. Rule Filtering:

- Set a confidence and/or lift threshold to filter out rules.
- Choose the rules that meet your business objectives. High-confidence rules are more reliable, while high-lift rules show stronger associations.

7.Interpretation and Action:

- Analyze the generated rules to gain insights into customer behavior.
- Use the insights to make data-driven decisions such as product placement, bundling, marketing, and pricing strategies.

8.Continuous Monitoring and Refinement:

- Regularly update and re-run the analysis as new transaction data becomes available.
- Adjust parameters and rules as needed based on changing business goals and market dynamics.

Market basket insights are a valuable tool for optimizing sales strategies, enhancing customer experience, and increasing revenue. The process helps businesses understand the relationships between products and make informed decisions to meet customer needs and boost profitability.

PROCEDURE:

FEATURE SELECTION:

Feature selection is an essential step in market basket insights analysis to identify which items or attributes are most relevant for discovering purchasing patterns and associations between products. Here's how you can perform feature selection for market basket insights:

- **Transaction-Level Data:**

The primary feature is the list of items in each transaction. Each unique item forms a potential feature. You may represent items as binary (0 or 1) features, indicating their presence or absence in a transaction.

- **Support Threshold:**

Define a minimum support threshold. This threshold helps filter out infrequent items, reducing the dimensionality of the dataset. Items with support below this threshold may be excluded from analysis.

- **Frequent Itemset Generation:**

Use algorithms like the Apriori algorithm to generate frequent itemsets. These itemsets are combinations of items that meet the defined support threshold. Frequent itemsets become the relevant features for market basket insights.

- **Association Rules:**

From the frequent itemsets, generate association rules. Each rule has an antecedent (the item(s) on the left-hand side) and a consequent (the item on the right-hand side). These rules become important features that capture item associations.

- **Confidence and Lift Threshold:**

Filter the association rules based on confidence and lift. Rules that don't meet these criteria may be excluded from further analysis.

- **PruningNon-Informative Features:**

After generating frequent itemsets and association rules, you can further refine the selection by pruning rules or items that are not informative or actionable for your business objectives.

- **Domain Knowledge:**

Consider domain-specific knowledge and business goals. Some items may have inherent importance due to their role in cross-selling or bundling, regardless of their support or lift values.

- **Iterative Process:**

Feature selection is often an iterative process. You can experiment with different support, confidence, and lift thresholds to identify the most relevant features for your specific analysis.

- **Evaluation Metrics:**

Consider using evaluation metrics such as lift, confidence, and conviction to assess the importance of rules and items. This can help in the final selection of features.

- **Visualization:**

Visualization techniques like heatmaps can assist in identifying item associations and selecting relevant features.

In market basket insights, the relevant features are the frequent itemsets and association rules, as they capture the relationships and patterns in customer purchasing behavior. Proper feature selection is crucial for a focused and meaningful analysis, allowing businesses to make informed decisions and optimize their strategies effectively.

MODEL TRAINING:

Training a model for market basket insights typically involves the use of data mining and machine learning techniques. Here's a high-level overview of the process:

Market basket analysis involves discovering relationships between items in a transactional dataset to gain insights into customer purchasing behavior. There are two primary types of rules generated in market basket analysis: association rules and sequence rules. Here's a brief definition and an example program for each type:

1.Association Rules:

Association rules identify relationships between items purchased together in a transaction. These rules are typically in the form of "if A, then B" and are evaluated based on metrics like support, confidence, and lift.

- **Example Program in Python using Apriori**:

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

# Sample transaction dataset
dataset = [
    ['milk', 'bread', 'nuts'],
    ['milk', 'bread', 'diapers'],
    ['milk', 'diapers'],
    ['milk', 'bread'],
    ['eggs', 'nuts']
]

df = pd.DataFrame(dataset, columns=['item1', 'item2', 'item3'])

# Use Apriori to find frequent itemsets
frequent_itemsets = apriori(df, min_support=0.4, use_colnames=True)

# Generate association rules
rules = association_rules(frequent_itemsets, metric='lift', min_threshold=1.0)
print(rules)
~~~

2.Sequence Rules:

Sequence rules focus on the order in which items are purchased and identify sequences of itemsets that frequently occur in transactions. These rules are used in scenarios like analyzing clickstreams or time-based data.

- **Example Program in Python using PrefixSpan**:

~~~
```python
from prefixspan import PrefixSpan

# Sample sequence dataset
sequences = [
    [1, 2, 3, 4],
    [1, 3, 2],
    [2, 4],
    [3, 1, 2, 4],
~~~

```
      [2, 3]
  ]

  # Use PrefixSpan to find frequent sequences
  ps = PrefixSpan(sequences)
  frequent_sequences = ps.frequent(2)

  print(frequent_sequences)
```

3.Data Collection :

```
python
import requests
from bs4 import BeautifulSoup

# Define the URL of the webpage you want to scrape
url = 'https://example.com'

# Send an HTTP GET request to the URL
response = requests.get(url)

# Check if the request was successful (status code 200)
if response.status_code == 200:
    # Parse the HTML content of the webpage using BeautifulSoup
    soup = BeautifulSoup(response.text, 'html.parser')

    # Extract data from the webpage by selecting elements using CSS selectors
    # For example, let's extract all the links (anchor tags) from the webpage
    links = soup.find_all('a')

    # Print the extracted links
    for link in links:
        print(link.get('href'))

else:
    print(f"Failed to retrieve data. Status code: {response.status_code}")
```

4.Model Evaluation :

Model evaluation is a critical step in assessing the performance of machine learning models. Here's a Python program that demonstrates how to evaluate a classification model using common evaluation metrics like accuracy, precision, recall, F1-score, and the confusion matrix:

```python
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression

# Load a sample dataset (you should replace this with your dataset)
data = load_iris()
X = data.data
y = (data.target == 2).astype(int)  # Binary classification (e.g., classifying whether an iris is a Virginica)

# Split the dataset into a training and testing set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a classification model (you should replace this with your model)
model = LogisticRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
confusion = confusion_matrix(y_test, y_pred)

# Print the evaluation metrics
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-Score: {f1:.2f}")
print("Confusion Matrix:")
print(confusion)
```

5.Data Preprocessing:

Data preprocessing is a crucial step in data analysis and machine learning. Here's a Python program that demonstrates common data preprocessing tasks, including handling missing values, encoding categorical variables, and splitting data into training and testing sets:

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# Sample dataset with missing values and categorical variables
data = {
    'Age': [25, 30, 35, None, 28, 22, 40],
    'Gender': ['Male', 'Female', 'Male', 'Male', 'Female', 'Female', 'Male'],
    'Salary': [50000, 60000, 75000, 60000, 80000, None, 90000],
    'Purchased': ['No', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes']
}

# Create a DataFrame from the data
df = pd.DataFrame(data)

# Handle missing values (e.g., fill with the mean for numerical columns)
df['Age'].fillna(df['Age'].mean(), inplace=True)
df['Salary'].fillna(df['Salary'].mean(), inplace=True)

# Encode categorical variables (e.g., 'Gender' and 'Purchased')
le = LabelEncoder()
df['Gender'] = le.fit_transform(df['Gender'])
df['Purchased'] = le.fit_transform(df['Purchased'])

# Split the dataset into training and testing sets
X = df.drop('Purchased', axis=1)  # Features
y = df['Purchased']  # Target variable
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Display the preprocessed data
print("Preprocessed DataFrame:")
print(df)
print("\nTraining Data:")
print(X_train)
print(y_train)
print("\nTesting Data:")
print(X_test)
print(y_test)
```

MODEL TRAINING:

Training a model for market basket insights typically involves using techniques like association rule mining, machine learning, or deep learning. Here's a high-level overview of the process:

1.Data Collection: Gather data on customer transactions, including what items were purchased together. This data can come from point-of-sale systems, e-commerce platforms, or other sources.

2.Data Preprocessing: Clean and preprocess the data. This may involve removing duplicates, handling missing values, and converting it into a suitable format for analysis.

3.Association Rule Mining:

- Use algorithms like Apriori or FP-Growth to discover frequent itemsets in the data.
- Generate association rules that reveal item relationships, typically in the form of "if-then" statements.

4.Machine Learning(Optional):

- You can use machine learning models like decision trees, random forests, or clustering algorithms to uncover patterns and associations in the data.
- Feature engineering can be beneficial to create relevant features for your model.

5.Deep Learning (Optional):

- For more complex and large-scale datasets, deep learning techniques like neural collaborative filtering can be employed to recommend products based on historical data.

6.Evaluation:

- Assess the quality and relevance of the discovered rules or the performance of your machine learning/deep learning model. Common metrics include support, confidence, lift, and F1-score.

7.Visualization:

- Create visualizations to present the market basket insights effectively, such as heatmaps, network graphs, or bar charts.

8.Deployment:

- Implement the model in a real-world setting, such as an e-commerce website, to provide product recommendations or other insights to customers.

9. Monitoring and Maintenance:

- Continuously monitor the model's performance and retrain it as new data becomes available to ensure that it stays accurate and relevant.

Remember that the choice of algorithms and techniques depends on the specific goals of your analysis and the characteristics of your dataset. The quality and quantity of data are crucial for obtaining meaningful market basket insights.

1.Dividing dataset in to features and target variable:

```
python
import pandas as pd

# Load your transaction data into a DataFrame
data = pd.read_csv("your_transaction_data.csv")

# Identify your target variable(s) - the item or combination you want to analyze
target_items = ["item_A", "item_B", "item_C"]

# Create a new DataFrame for features
features = data.drop(target_items, axis=1)

# Create a new DataFrame for the target variable
target = data[target_items]

# Now, 'features' contains all the items purchased for each transaction,
# and 'target' contains the items you want to analyze for market basket insights.
```

````
# You can proceed with your market basket analysis using these DataFrames.
````

2.Split the data into training and test sets:

~~~
```python
from sklearn.model_selection import train_test_split

# Assuming you already have 'features' and 'target' DataFrames from the previous example

# Split the data into training and test sets (e.g., 80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

# 'X_train' and 'y_train' will contain the training features and target variables.
# 'X_test' and 'y_test' will contain the test features and target variables.

# You can use these datasets to build and evaluate your market basket analysis models.
~~~

3.Train the model on the test set :

It's not typical to train a model on the test set in machine learning. Instead, you should train your model on the training set and then evaluate it on the test set to assess its performance.

4.Evaluate the model on the test set:

To evaluate a market basket analysis model on the test set, you need to use appropriate metrics that are relevant to your specific analysis goals. Common metrics for market basket analysis include lift, support, confidence, and others.

MODEL EVALUATION:

In market basket analysis, there are several common metrics used to evaluate the performance of models. The choice of metrics depends on the specific goals of your analysis. Here are some commonly used metrics for evaluating market basket insights:

****Support:**** Support measures how frequently an itemset (combination of items) appears in the dataset. Higher support indicates that the itemset is more common.

Confidence: Confidence measures the likelihood that if item A is purchased, item B will also be purchased. It's calculated as the support for the combination of items A and B divided by the support for item A.

Lift: Lift is a measure of how much more likely item B is to be purchased when item A is purchased, compared to when item B is purchased without any knowledge of item A. It helps identify whether there is a true association between the items.

Leverage: Leverage measures the difference between the observed frequency of item A and B appearing together and what would be expected if they were independent. A higher positive leverage indicates a positive association.

Conviction: Conviction measures the dependency between items A and B. A higher conviction value indicates a stronger association between the items.

Apriori Algorithm Metrics: If you're using the Apriori algorithm, you might consider metrics like the number of frequent itemsets, association rules, and their respective measures (support, confidence, lift).

Evaluation of predicted data :

~~~
```python
# Import necessary libraries
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

# Load your transaction data into a DataFrame (e.g., using Pandas)
# Replace 'your_data.csv' with the actual file path or data source
import pandas as pd
data = pd.read_csv('your_data.csv')

# Data preprocessing (cleaning and formatting as needed)

# Perform one-hot encoding to convert items to binary values
basket_sets = pd.get_dummies(data, columns=['Item'])

# Apply the Apriori algorithm to find frequent item sets
frequent_itemsets = apriori(basket_sets, min_support=0.05, use_colnames=True)
~~~

```
# Generate association rules
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.0)

# Filter and rank rules based on criteria
filtered_rules = rules[(rules['confidence'] > 0.5) & (rules['lift'] > 1.0)]

# Display the resulting association rules
print(filtered_rules)
```

FEATURES ENGINEERING:

Feature engineering for market basket insights involves creating meaningful features from your transaction data to help identify patterns and associations between items. Here are some feature engineering techniques:

1.Binary Encoding: Create binary features for each item indicating whether it was bought in a transaction (1 for bought, 0 for not bought).

2.Item Frequency: Calculate the frequency of each item's occurrence in the dataset. This can help identify popular or rare items.

3.Item Co-occurrence: Create features that represent the co-occurrence of pairs or groups of items in transactions. For example, you can count how often items A and B are purchased together.

4.Transaction Count: Feature that represents the total number of items in a transaction. This can help in understanding the size of transactions.

5.Time-based Features: If your data includes timestamps, create features related to time, such as the time of day, day of the week, or season, to identify temporal patterns.
6.Customer Segmentation: Group customers based on their shopping behavior. Features can include segment membership, average transaction size, or frequency of purchases.

7.Basket Diversity: Calculate a diversity metric to represent how varied or similar items are within a basket. It can be helpful in understanding customer preferences.

8.Sequential Patterns: If your data includes sequences of transactions (e.g., website clickstreams), create features to identify sequential patterns, like the order in which items are added to the basket.

9.Loyalty Features: Features related to customer loyalty, such as the number of repeat purchases or the time since the last purchase.

10.Item Categories: If items have categories or tags, create features based on these categories to explore relationships between product categories.

11.Geographic Features: If your data includes location information, consider features related to geographic patterns, like store location or customer location.

12.Text Analysis: If you have item descriptions or customer comments, perform text analysis to extract relevant keywords or sentiments that can be used as features.

13.Market Basket Analysis Metrics: Features derived from market basket analysis results, such as item support, confidence, lift, or association rules.

Remember that the choice of features will depend on your specific dataset, business objectives, and the analysis techniques you plan to use. Effective feature engineering can significantly enhance the quality of your market basket insights.

CONCLUSION:

In the context of market basket insights, the conclusion serves as a critical component of the analysis, summarizing key findings, implications, and recommendations.

Market basket insights reveal product affinities, aid in customer segmentation, optimize inventory, inform pricing and promotions, enable personalization, and highlight seasonal trends. Businesses use this data to enhance operations, drive sales, and improve the customer experience.

Team Members

MohamedRiyasudeen M

Tamil Selvan T

Sirajdeen M

Sathishkumar N

Srithirumalai G

JP College Of Engineering