|  |
| --- |
| **Course: ARTIFICIAL INTELLIGENCE**  **TITLE: MARKET BASKET INSIGHTS**  **PHASE 1 SUBMISSION** |
|  |
|  |

|  |
| --- |
| TEAM MEMBERS:  MOHANA PRIYA .N (au 311521106061)  MOUNICKA G.G (au 311521106062  PADMA PRIYA .M. (au 311521106069)  YAMINI.H( au 311521106114) |

ABSTRACT

Market basket analysis (MBA) is a data mining technique that is used to discover relationships between items that are frequently purchased together. It is a powerful tool that can be used by retailers to understand customer purchase patterns and make more informed decisions about product placement, promotions, and inventory management.

**Module**

The following is a high-level overview of a module for market basket analysis for groceries:

Data preparation: The first step is to prepare the grocery transaction data. This may involve cleaning the data, removing outliers, and encoding the data into a format that is compatible with the MBA algorithm. Association rule mining: The next step is to mine the transaction data for association rules. Association rules are of the form:

IF antecedent THEN consequent where antecedent and consequent are sets of items. The antecedent is the set of items that are purchased together frequently, and the consequent is the set of items that are likely to be purchased when the antecedent items are purchased.There are a number of different algorithms that can be used to mine association rules. One of the most common algorithms is the Apriori algorithm. The minimum confidence parameter specifies the minimum percentage of transactions that contain the antecedent items that must also contain the consequent items in order for the association rule to be considered valid.

Analysis: Once the association rules have been mined, they can be analyzed to identify patterns and trends. For example, a retailer may find that customers who purchase diapers are also likely to purchase baby wipes and formula. This information can then be used to develop targeted promotions or to place these products together in the store.

Benefits

Market basket analysis can provide a number of benefits to retailers, including:

Increased sales: Retailers can use MBA to identify patterns and trends in customer purchase behavior. This information can then be used to develop targeted promotions or to place products together in the store in a way that encourages customers to purchase more items.

Reduced inventory costs: Retailers can use MBA to identify items that are frequently purchased together. This information can then be used to ensure that these items are always in stock, while reducing the amount of inventory that is held for items that are not purchased as frequently.

Improved customer satisfaction: Retailers can use MBA to understand their customers' needs and preferences better. This information can then be used to improve the customer experience, such as by offering more convenient shopping options or by providing more personalized recommendations.

**Dataset:**

• The Online Retail II data set, which includes the sales data of the UK-based online sales store, was used.

• Sales data between 01/12/2009 - 09/12/2011 are included in the data set.

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

A screenshot of a computer screen

Description automatically generated **Problem Statements:**

Suggesting products to users at the basket stage. In this study, we will apply Market Basket analysis using the Apriori algorithm:

**1.** Import Data & Data Preprocessing

**2.** Preparing Invoice-Product Matrix fot ARL Data Structure

**3.** Determination of Association Rules

**4.** Suggesting appropriate product offers to customers at the basket stage

**5.** Functionalization

**Program**

df\_ = pd.read\_excel('../input/online-retail-dataset/online\_retail\_II.xlsx', sheet\_name='Year 2010-2011')

*# Import & Filter Data:*

def data\_filter(dataframe, country=False, Country=""):

if country:

dataframe = dataframe[dataframe["Country"] == Country]

return dataframe

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 data\_prep(dataframe):

*# Data preprocessing:*

dataframe.dropna(inplace=True)

*# Delete if the product name contains "POST":*

dataframe = dataframe[~dataframe["StockCode"].str.contains("POST", na=False)]

dataframe = dataframe[~dataframe["Invoice"].str.contains("C", na=False)]

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

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

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

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

return dataframe

*# Invoice Product Matrix:*

def create\_invoice\_product\_df(dataframe, id=False):

if id:

return dataframe.groupby(['Invoice', "StockCode"])['Quantity'].sum().unstack().fillna(0). \

applymap(lambda x: 1 if x > 0 else 0)

else:

return dataframe.groupby(['Invoice', 'Description'])['Quantity'].sum().unstack().fillna(0). \

applymap(lambda x: 1 if x > 0 else 0)

*# Find Product name with Stock Code:*

def check\_id(dataframe, stockcode):

product\_name = dataframe[dataframe["StockCode"] == stockcode]["Description"].unique()[0]

return stockcode, product\_name

*# Apriori Algorithm & ARL Rules:*

def apriori\_alg(dataframe, support\_val=0.01):

inv\_pro\_df = create\_invoice\_product\_df(dataframe, id=True)

frequent\_itemsets = apriori(inv\_pro\_df, min\_support=support\_val, use\_colnames=True)

rules = association\_rules(frequent\_itemsets, metric="support", min\_threshold=support\_val)

sorted\_rules = rules.sort\_values("support", ascending=False)

return sorted\_rules

def recommend\_product(dataframe, product\_id, support\_val= 0.01, num\_of\_products=5):

sorted\_rules = apriori\_alg(dataframe, support\_val)

recommendation\_list = []

for idx, product **in** enumerate(sorted\_rules["antecedents"]):

for j **in** list(product):

if j == product\_id:

recommendation\_list.append(list(sorted\_rules.iloc[idx]["consequents"])[0])

recommendation\_list = list( dict.fromkeys(recommendation\_list) )

return(recommendation\_list[0:num\_of\_products])

*# Data Preparation:*

df = df\_.copy()

df = data\_prep(df)

df = data\_filter(df,country=True,Country="Germany")

df.head()

**Output**



**Recommendation System Function**

**Program**

def recommendation\_system\_func(dataframe,support\_val=0.01, num\_of\_products= 5 ):

product\_id = input("Enter a product id:")

if product\_id **in** list(dataframe["StockCode"].astype("str").unique()):

product\_list = recommend\_product(dataframe, int(product\_id), support\_val, num\_of\_products)

if len(product\_list) == 0:

print("There is no product can be recommended!")

else:

print("Related products with product id -" , product\_id , "can be seen below:")

for i **in** range(0, len(product\_list[0:num\_of\_products])):

print(check\_id(dataframe, product\_list[i]))

else:

print("Invalid Product Id, try again!")

**Output**

*# Enter product id - 2 (22725)*

Related products with product id - 22725 can be seen below:

(22727, 'ALARM CLOCK BAKELIKE RED ')

(22326, 'ROUND SNACK BOXES SET OF4 WOODLAND ')

(22728, 'ALARM CLOCK BAKELIKE PINK')

(22726, 'ALARM CLOCK BAKELIKE GREEN')

(22729, 'ALARM CLOCK BAKELIKE ORANGE')

recommendation\_system\_func(df)

There is no product can be recommended!

recommendation\_system\_func(df)

Invalid Product Id, try again!

**Conclusion**

Market basket analysis is a powerful tool that can be used by retailers to understand customer purchase patterns and make more informed decisions about product placement, promotions, and inventory management. By analyzing the items that customers frequently purchase together, retailers can develop strategies to increase sales, reduce inventory costs, and improve customer satisfaction.