****

**Assesment Report**

on

**“Market Analysis”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

By

Shikhar Maheshwari (202401100300230)

**Under the supervision of**

“Mr. Abhishek Shukla Sir”

**KIET Group of Institutions, Ghaziabad**

Affiliated to

**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

(Formerly UPTU)

**April, 2025**

**Introduction**

Market Basket Analysis (MBA) is a data mining technique used to uncover relationships between items frequently purchased together. It plays a vital role in understanding customer purchasing behavior and supporting business strategies such as product placement, bundling, and targeted marketing.

In this project, we simulate customer transactions based on real-world retail data and apply the Apriori algorithm to discover frequent itemsets and generate association rules. Additionally, we classify customers into high and low spenders using a logistic regression model and perform clustering to segment customers based on their shopping patterns. These insights help in making data-driven decisions to improve customer engagement and sales.

**Methodology**

The project follows a structured approach to perform Market Basket Analysis using simulated transaction data and machine learning techniques:

**1. Data Preparation**

We begin by uploading a real-world dataset and randomly selecting a subset of aisle names. Using these, we simulate 500 customer transactions with varying item counts. Customers purchasing more than 4 items are labeled as high spenders, while others are labeled as low spenders.

**2. Association Rule Mining**

The transaction data is one-hot encoded using TransactionEncoder. The **Apriori algorithm** is applied to identify frequent itemsets with a minimum support of 0.05. From these, **association rules** are generated using a confidence threshold of 0.3, helping us uncover meaningful item relationships.

**3. Customer Classification**

We use the number of items in a transaction as a feature to train a **logistic regression** model that predicts whether a customer is a high or low spender. The model’s performance is evaluated using accuracy, precision, recall, and a confusion matrix.

**4. Customer Segmentation**

To understand different customer profiles, we apply **K-Means clustering** on the one-hot encoded transaction data. **PCA (Principal Component Analysis)** is used to reduce dimensions and visualize customer clusters based on their purchase behavior.

**CODE**

# STEP 1: Load and Simulate Transaction Data

import pandas as pd

import numpy as np

import random

import seaborn as sns

import matplotlib.pyplot as plt

from google.colab import files

uploaded = files.upload() # Upload your "10. Market Basket Analysis.csv"

df\_aisles = pd.read\_csv("10. Market Basket Analysis.csv")

aisles = df\_aisles['aisle'].sample(20, random\_state=42).tolist()

transactions = []

customer\_labels = []

np.random.seed(42)

for \_ in range(500):

num\_items = np.random.randint(1, 8)

items = random.sample(aisles, num\_items)

transactions.append(items)

customer\_labels.append(1 if num\_items > 4 else 0) # High spender if more than 4 items

# STEP 2: Association Rule Mining (Apriori)

!pip install mlxtend

from mlxtend.preprocessing import TransactionEncoder

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

te = TransactionEncoder()

te\_array = te.fit(transactions).transform(transactions)

df\_trans = pd.DataFrame(te\_array, columns=te.columns\_)

frequent\_itemsets = apriori(df\_trans, min\_support=0.05, use\_colnames=True)

rules = association\_rules(frequent\_itemsets, metric="confidence", min\_threshold=0.3)

print("Top 5 Association Rules:")

display(rules.sort\_values(by='confidence', ascending=False).head())

# STEP 3: Classification (High vs. Low Spender)

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix, accuracy\_score, precision\_score, recall\_score

X = np.array([len(t) for t in transactions]).reshape(-1, 1)

y = np.array(customer\_labels)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LogisticRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

cm = confusion\_matrix(y\_test, y\_pred)

acc = accuracy\_score(y\_test, y\_pred)

prec = precision\_score(y\_test, y\_pred)

rec = recall\_score(y\_test, y\_pred)

plt.figure(figsize=(6, 4))

sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=["Low", "High"], yticklabels=["Low", "High"])

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.title("Confusion Matrix")

plt.show()

print(f"Accuracy: {acc:.2f}")

print(f"Precision: {prec:.2f}")

print(f"Recall: {rec:.2f}")

# STEP 4: Clustering and Customer Segmentation

from sklearn.cluster import KMeans

from sklearn.decomposition import PCA

item\_df = pd.DataFrame(te\_array.astype(int), columns=te.columns\_)

kmeans = KMeans(n\_clusters=3, random\_state=42)

clusters = kmeans.fit\_predict(item\_df)

reduced = PCA(n\_components=2).fit\_transform(item\_df)

plt.figure(figsize=(8, 6))

sns.scatterplot(x=reduced[:, 0], y=reduced[:, 1], hue=clusters, palette="Set2")

plt.title("Customer Segmentation Based on Aisle Preferences")

plt.xlabel("PCA 1")

plt.ylabel("PCA 2")

plt.show()

**OUTPUT**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a graph

Description automatically generated**

**A diagram of a customer segmentation

Description automatically generated**

**References/Credits**

.

 Mlxtend Documentation

 Scikit-learn Documentation

 Dataset: Market Basket Analysis

 Google Colab