Market Basket Insights using Machine Learning

Project Title: Market Basket Insights

Phase 5: Final submission.

Market Basket Insights

Problem definition:

The problem is to perform market basket analysis on a provided dataset to unveil hidden patterns and associations between products. The goal is to understand customer purchasing behavior and identify potential cross-selling opportunities for a retail business. This project involves using association analysis techniques, such as Apriori algorithm, to find frequently co-occurring products and generate insights for business optimization.

Design thinking:

1.Data Source: Choose a dataset containing transaction data, including lists of purchased products.

2.Data Preprocessing: Prepare the transaction data by transforming it into a suitable format for association analysis.

3.Association Analysis: Utilize the Apriori algorithm to identify frequent itemsets and generate association rules.

4.Insights Generation: Interpret the association rules to understand customer behavior and cross-selling opportunities.

5.Visualization: Create visualizations to present the discovered associations and insights.

6.Business Recommendations: Provide actionable recommendations for the retail business based on the insights.

Innovation:

1) Target Audience: Identify your target audience and their preferences. Tailor the contents of the basket to suit their tastes and needs.
2) Occasion and Theme: Determine the occasion or theme for the basket, whether it's a holiday, corporate event, or a specific promotion.
3) Branding: Incorporate your company's branding elements such as logos, colors, and custom packaging to reinforce your brand identity.
4) High-Quality Products: Include high-quality products or items in the basket to create a positive impression and provide value to the recipient.
5) Personalization: Whenever possible,personalize the baskets with the recipient's name or a personal message to make it more meaningful.
6) Variety: Offer a variety of items within the basket to cater to different tastes and preferences.
7) Presentation: Pay attention to the presentation, including how the items are arranged within the basket. Presentation can greatly impact the perceived value.
8) Budget: Determine a budget for your marketing baskets and ensure it aligns with your marketing goals.
9) Distribution: Plan how you will distribute the baskets, whether it's through direct mail, in-person delivery, or at an event.
10) Measurable Goals: Set specific goals for your marketing campaign using these baskets,whether it's increased brand awareness,customer acquisition, or loyalty.
11) Follow-Up: After sending out the baskets,follow up with recipients to gather feedback and assess the impact of your campaign.
12) Legal Considerations: Be aware of any legal regulations or restrictions related to sending promotional gifts or baskets in your region or industry.

Necessary step to follow:

1.Import Libraries:

Start by importing the necessary libraries.

Program:

```
import pandas as pd

# Load the dataset

dataset_path = '/kaggle/input/market-basket-analysis/Assignment-1_Data.xlsx'

df = pd.read_excel(dataset_path)
```

2.Initial Exploration:

We'll perform an initial exploration of the dataset to understand its structure and characteristics.

Program:

```
print("Number of rows and columns:", df.shape)
print("\nData Types and Missing Values:")
print(df.info())
print("\nFirst few rows of the dataset:")
print(df.head())
```

Output:

```
Number of rows and columns: (522064, 7)
Data Types and Missing Values:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 522064 entries, 0 to 522063
Data columns (total 7 columns):
# Column Non-Null Count Dtype
--- ------ -------------- -----
0 BillNo 522064 non-null object
1 Itemname 520609 non-null object
2 Quantity 522064 non-null int64
3 Date 522064 non-null datetime64[ns]
4 Price 522064 non-null float64
5 CustomerID 388023 non-null float64
6 Country 522064 non-null object
Dtypes: datetime64[ns](1), float64(2), int64(1), object(3)
Memory usage: 27.9+ MB
None
First few rows of the dataset:
BillNo       Itemname   Quantity  Date \
```

0 536365 WHITE HANGING HEART T-LIGHT HOLDER 6 2010-12-00

08:26:00

1 536365 WHITE METAL LANTERN 6 2010-12-01 08:26:00

2 536365 CREAM CUPID HEARTS COAT HANGER 8 2010-12-01

08:26:00

3 536365 KNITTED UNION FLAG HOT WATER BOTTLE 6 2010-12-01

08:26:00

4 536365 RED WOOLLY HOTTIE WHITE HEART. 6 2010-12-01

08:26:00

Price CustomerID Country

0 2.55 17850.0 United Kingdom

1 3.39 17850.0 United Kingdom

2 2.75 17850.0 United Kingdom

3 3.39 17850.0 United Kingdom

Preprocessing:

We'll preprocess the data to ensure it's ready for analysis.

Program:

```
#Check Missing Values
print("Missing Values:")
print(df.isnull().sum())
#Drop Rows with Missing Values
df.dropna(inplace=True)
```

Output:

Missing Values:

BillNo 0

Itemname 1455

Quantity 0

Date 0

Price 0

CustomerID 134041

Country 0

dtype: int64n

DataFrame:

Program:

```
# Convert dataframe into transaction data
Transaction_data=df.groupby(['BillNo','Date'])['Itemname'].apply(lambda
X: ', '.join(x)).reset_index()
#Drop Unnecessary Columns
columns_to_drop = ['BillNo', 'Date']
transaction_data.drop(columns=columns_to_drop, inplace=True)
# Save the transaction data to a CSV file
Transaction_data_path'/kaggle/working/transaction_data.csv'transaction_data.to_csv(transaction_data
_path, index=False)
# Display the first few rows of the transaction data
print("\nTransaction Data for Association Rule Mining:")
print(transaction_data.head())
transaction_data.shape
```

Output:

```
Transaction Data for Association Rule Mining:
                Itemname
0 WHITE HANGING HEART T-LIGHT HOLDER, WHITE META…
1 HAND WARMER UNION JACK, HAND WARMER RED POLKA DOT
2 ASSORTED COLOUR BIRD ORNAMENT, POPPY'S PLAYHOU…
3 JAM MAKING SET WITH JARS, RED COAT RACK PARIS…
```

4 BATH BUILDING BLOCK WORD

(18192, 1)

FEATURE ENGINEERING:

Feature engineering typically involve creating new features or transforming existing once to improve the performance of a machine learning model this specific code for feature Engineering can vary widely depending on data set .

Some of the main reasons include:

1.Improve User Experience: The primary reason we engineer features is to enhance the user experience of a product or service. By adding new features, we can make the product more intuitive, efficient, and user-friendly, which can increase user satisfaction and engagement.

2.Competitive Advantage: Another reason we engineer features is to gain a competitive advantage in the marketplace. By offering unique and innovative features, we can differentiate our product from competitors and attract more customers.

3.Meet Customer Needs: We engineer features to meet the evolving needs of customers. By analyzing user feedback, market trends, and customer behavior, we can identify areas where new features could enhance the product's value and meet customer needs

4.Increase Revenue: Features can also be engineered to generate more revenue. For example, a new feature that streamlines the checkout process can increase sales, or a feature that provides additional functionality could lead to more upsells or cross-sells.

5.Future-Proofing: Engineering features can also be done to future-proof a product or service. By anticipating future trends and potential customer needs, we can develop features that ensure the product remains relevant and useful in the long term.

DATA COLLECTION :

Gather transaction data that includes information on what items were purchased together. This can be obtained from point-of-sales system or e-commerce platforms.

Data collection is a systematic process of gathering observations or measurements. Whether you are performing research for business, governmental or academic purposes, data collection allows you to gain first-hand knowledge and original insights into your research while methods and aims may differ between fields, the overall process of data collection remains largely the same. Before we begin collecting data, you need to consider

- The aim of the research
- The type of data that you will collect
- The methods and procedures you will use to collect, store, and process the data

```
import pandas as pd
data={
    "BillNo':[536365,536366,536367.....]
    "Itemname':["WHITE HANGING HEART T-LIGHT HOLDER","WHITE METAL LATERN","CREAM CUPIDS HEARTS COAT HANGER"...)]
    "Quantity':[6,6,8.....]
    "Date':[########,########,########....]
    "Price':[2.55,3.39,2.75....]
    "CustomerID':[17850,1850,17850....]
    "Country':[United Kingdom,United Kingdom,United Kingdom.....]
}
```

FEATURE ENGINEERING PROCESS:

We engineer features to improve the performance of machine learning models by providing them with relevant and informative input data. Raw data may contain noise, irrelevant information, or missing values, which can lead to inaccurate or biased model predictions. By engineering features, we can extract meaningful information from the raw data, create new variables that capture important patterns and relationships, and transform the data into a more suitable format for machine learning algorithms.

Feature engineering can also help in addressing issues such as overfitting, underfitting, and high dimensionality. For example, by reducing the number of features, we can prevent the model from becoming too complex or overfitting to the training data. By selecting the most relevant features, we can improve the model's accuracy and interpretability.

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import OneHotEncoder
```

```
data=pd.read_csv("Assignment-1_Data.csv")
```

```
numeric_features = ['bill no','date', 'customer id']
scaler = StandardScaler()
data[numeric_features] = scaler.fit_transform(data[numeric_features])
```

```
categorical_features = ['']
encoder = OneHotEncoder()
encoded_features = encoder.fit_transform(data[categorical_features]).toarray()
encoded_feature_names = encoder.get_feature_names(categorical_features)
data = pd.concat([data, pd.DataFrame(encoded_features, columns=encoded_feature_names)], axis=1)
data.drop(categorical_features, axis=1, inplace=True)
```

In addition, feature engineering is a crucial step in preparing data for analysis and decision-making in various fields, such as finance, healthcare, marketing, and social sciences. It can help uncover hiddeninsights, identify trends and patterns, and support data-driven decision-making.

Visualization :

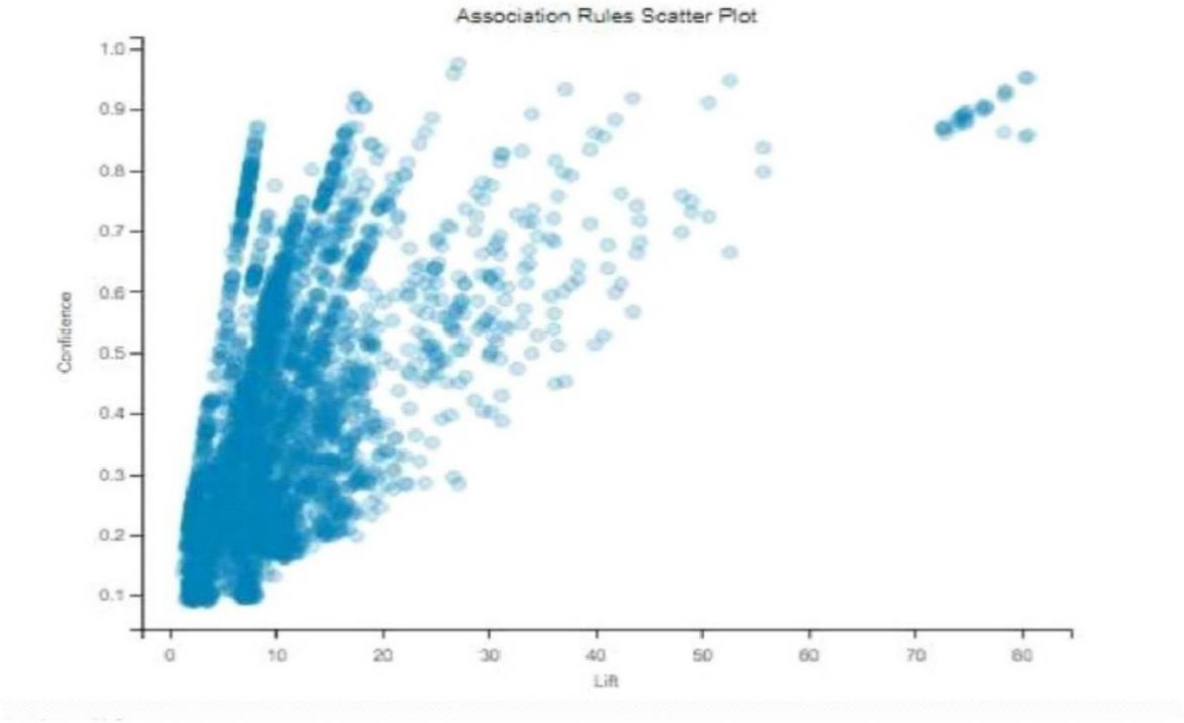
Data visualization is the representation of data through use of common graphics, such as charts, plots, Infographics, and even animations. These visual displays of information communicate complex data relationships and data-driven insights.

Types of Data Visualizations:

- Line charts and area charts: These visuals show change in one or more quantities by plotting a series of data points over time and are frequently used within predictive analytics. Line graphs utilize lines to demonstrate these changes while area charts connect data points with line segments, stacking variables on top of one another and using color to distinguish between variables.
- Histograms: This graph plots a distribution of numbers using a bar chart (with no spaces between the bars), representing the quantity of data that falls within a particular range. This visual makes it easy for an end user to identify outliers within a given dataset.
- Tables: This consists of rows and columns used to compare variables. Tables can show a great deal of information in a structured way, but they can also overwhelm users that are simply looking for high-level trends.
- Pie charts and stacked bar charts: These graphs are divided into sections that represent parts of a whole. They provide a simple way to organize data and compare the size of each component to one other.
- Scatter plots: These visuals are beneficial in reveling the relationship between two variables, and they are commonly used within regression data analysis. However, these can sometimes be confused with Bubble charts, which are used to visualize three variables via the x-axis, the y-axis, and the size of the Bubble.
- Heat maps: These graphical representation displays are helpful in visualizing behavioral data by location.This can be a location on a map, or even a webpage. Tree maps, which display hierarchical data as a set of nested shapes, typically rectangles. Treemaps are great for comparing the proportions between categories via their area size

OPEN SOURCE TOOLS:

```
import matplotlib.pyplot as plt
x=[0,10,20,30,40,50,60,70,80]
y=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
plt.scatter(x,y)
plt.xlabel("X AXIS LABEL")
plt.yaxis("Y AXIS LABEL")
plt.title("Scatter Plot")
plt.show()
```



Vega defines itself as “visualization grammar,” providing support to customize visualizations across large datasets which are accessible from the web.

Evaluation:

It is the performance of machine learning models is crucial for understanding how well they’re doing here’s is the basic example of how to evaluate a classification model using python and Scikit-learn

Perspectives :

The word “evaluation” has various connotations for different people, raising issues related to this Process that include,what type of evaluation should be conducted; why there should be an evaluation process and how the evaluation is integrated into a program, for the purpose of gaining greater knowledge and awareness? There are also various factors inherent in the evaluation process, for example to critically examine influences within a program that involve the gathering and analyzing of relative information about a program.

- Activities
- Characteristics
- Outcomes
- The making of judgments on a program
- Improving its effectiveness,
- Informed programming decisions

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.ensemble import RandomForestClassifier
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
# Print the results
print(f'Accuracy: {accuracy:.2f}')
print('Confusion Matrix:')
print(conf_matrix)
print('Classification Report:')
print(class_report)
```

Conclusion:

- ✓ Market Basket Analysis is a valuable technique that offers profound insights into customer behavior and product associations, making it an indispensable tool for businesses across various industries. In conclusion, here are some key takeaways.
- ✓ Enhanced Customer Understanding: MBA provides a deeper understanding of customer preferences and purchase patterns. By identifying which products are frequently bought together, businesses can tailor their strategies to meet customer demands more effectively.

- ✓ Improved Inventory Management: MBA helps businesses optimize inventory levels by stocking products that are commonly bought together. This not only reduces the risk of overstocking but also ensures that popular items are consistently available. Personalized Recommendations: Through the insights gained from market basket analysis, businesses can offer personalized product recommend