IBM PROJECT SUBMISSION

NAME: Samyuktha S R

ROLL NO: 422521104032

TITLE: Market Basket Insights

DOMAIN: Artificial Intelligence(AI)

COLLEGE NAME: University College of Engineering Villupuram.

COLLEGE CODE: 4225

PHASE 3 – DEVELOPMENT PART 1

MARKET BASKET INSIGHTS

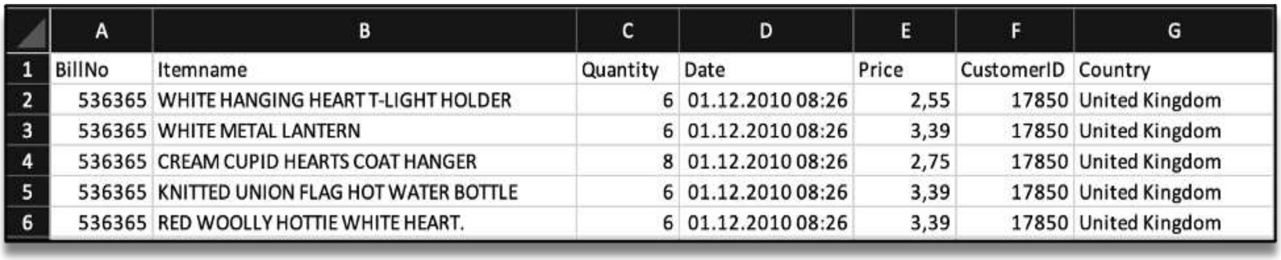
- Market basket insights are the findings from market basket analysis, a data mining technique that identifies patterns and associations between products frequently purchased together.
- By analyzing transactional data, such as customer purchase history or shopping cart contents, businesses can uncover hidden relationships between products and gain valuable insights into customer behavior.

DATA SET

The link for the chosen dataset is attached below:

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

- The attributes for the selected dataset are shown,

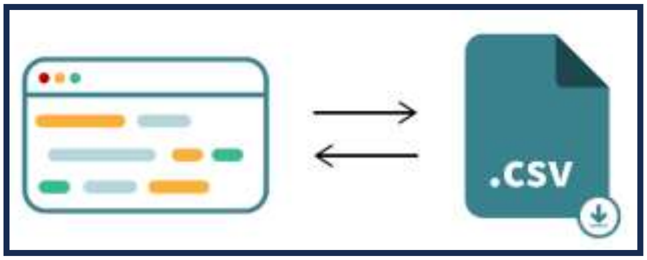


	A	B	C	D	E	F	G
1	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
2	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	01.12.2010 08:26	2,55	17850	United Kingdom
3	536365	WHITE METAL LANTERN	6	01.12.2010 08:26	3,39	17850	United Kingdom
4	536365	CREAM CUPID HEARTS COAT HANGER	8	01.12.2010 08:26	2,75	17850	United Kingdom
5	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01.12.2010 08:26	3,39	17850	United Kingdom
6	536365	RED WOOLLY HOTTIE WHITE HEART.	6	01.12.2010 08:26	3,39	17850	United Kingdom

BUILDING THE PROJECT

1. DATA LOADING:

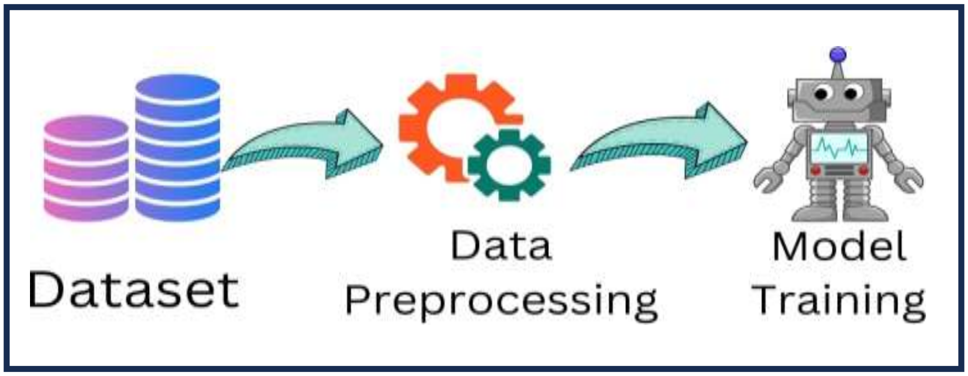
- Data loading refers to the process of importing data from one or more sources into a database, data warehouse, or other data storage system.



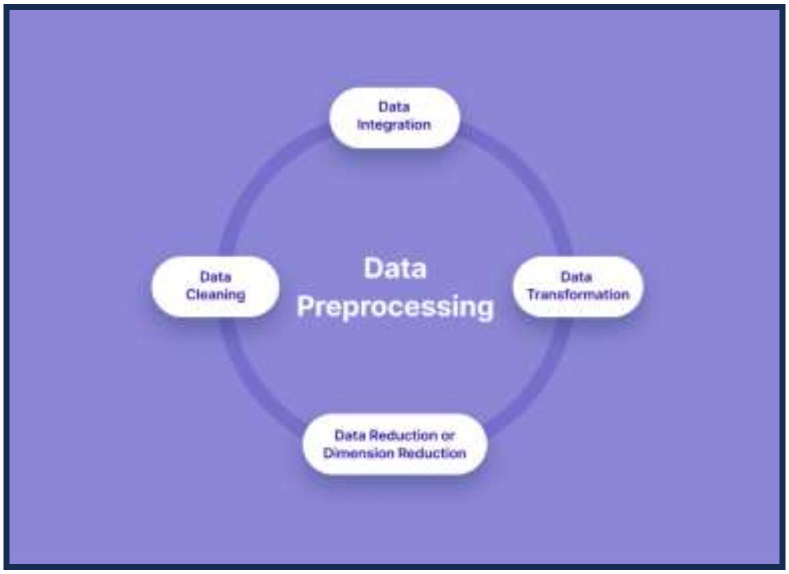
- This process involves extracting data from the source system, transforming it into a format suitable for the target system, and then loading it into the target system.
- Data loading can be performed on a regular basis (e.g., daily, weekly, monthly) to ensure that the target system is up-to-date with the latest data.

2. DATA PREPROCESSING:

Data preprocessing can be defined as the process of transforming raw data into a form that can be easily understood and analyzed by a machine learning algorithm. Data preprocessing involves various steps such as removing irrelevant data, dealing with missing values, dealing with outliers, scaling the data, and encoding categorical variables.



The following are the basic steps involved in data preprocessing:



(i).Data cleaning:

The process of detecting and correcting (or removing) invalid or irrelevant records from the dataset.

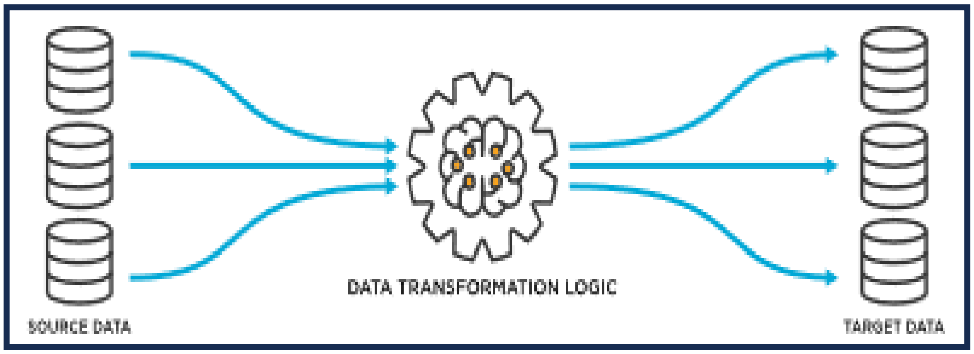
- ✓ Removal of Unwanted Observations.
- ✓ Managing Unwanted Outliers.
- ✓ Fixing Structural Error
- ✓ Handling Structural Data.

Cleaning the data:

- o Identify the data quality problems
- o Prioritize the data quality problems.
- o Validate the data.

(ii) Data integration:

Merging multiple datasets into one for analysis.



(iii) Data transformation:

The process of converting data from one form to another.

(iv) Data reduction:

- The process of reducing the amount of data by aggregating it or selecting a subset of relevant features.
- The process of converting continuous variables into categorical variables by dividing them into intervals.
- The process of scaling the features or attributes of a dataset to the same range to avoid the dominance of any particular feature.
- Data preprocessing is essential to ensure that the data is accurate, complete, and suitable for machine learning algorithms to produce accurate and reliable results.

Coding:

(1)LOADING

#Loading the dataset

```
data=pd.read_csv('/content/Assignment-1_DataN.csv')

data.head()   #viewing the data
```

#importing the necessary libraries

```
import pandas as pd

from sklearn.preprocessing import MinMaxScaler

import numpy as np

from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
```

#Loading the dataset

```
data=pd.read_csv('/content/Assignment-1_DataN.csv')

data.head()   #viewing the data
```

output

```
<ipython-input-7-3fa16f8c979c>:1: DtypeWarning: Columns (0) have mixed types. Specify
dtype option on import or set low_memory=False.
  data=pd.read_csv('/content/Assignment-1_DataN.csv')
     BillNo  Quantity  Price  CustomerID

0    536365         6   2.55     17850.0

1    536365         6   3.39     17850.0

2    536365         8   2.75     17850.0

3    536365         6   3.39     17850.0
```

```
   BillNo  Quantity  Price  CustomerID
4  536365         6   3.39     17850.0
```

```
data.tail()     #Viewing the end of the dataset
```

output

```
        BillNo  Quantity  Price  CustomerID
522059  581587        12   0.85     12680.0
522060  581587         6   2.10     12680.0
522061  581587         4   4.15     12680.0
522062  581587         4   4.15     12680.0
522063  581587         3   4.95     12680.0
```

```
#information about dataset
```

```
data.info()
```

output

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 522064 entries, 0 to 522063
Data columns (total 4 columns):
 #   Column      Non-Null Count   Dtype  
---  ------      --------------   -----  
 0   BillNo      522064 non-null  object 
 1   Quantity    522064 non-null  int64  
 2   Price       522064 non-null  float64
 3   CustomerID  388023 non-null  float64
dtypes: float64(2), int64(1), object(1)
memory usage: 15.9+ MB
CodeText
```

```
#Counting the number of Data
```

```
data.count()
```

output

```
 BillNo 522064
Quantity 522064
Price 522064
CustomerID 388023
dtype: int64
```

#Printing the attribute

```
data.BillNo
```

output

```
  0 536365
 1 536365
2 536365
3 536365
4 536365
...
522059 581587
522060 581587
522061 581587
522062 581587
522063 581587
Name: BillNo, Length: 522064, dtype: object
```

#type of the data

```
type(data)
```

output

```
pandas.core.frame.DataFrame
```

#printing the shape

```
data.shape
```

output

```
(419475, 4)
```

(2)PRE-PROCESSING

(i) Cleaning

```
#Handling Missing Data.
data['Quantity'].fillna(data['Quantity'].mean(),inplace=True)


data['Price'].fillna(data['Price'].mean(),inplace=True)
```

```
#Removes the null value

print(data.isnull().sum())
```

output

```
BillNo         0
Quantity       0
Price          0
CustomerID  40749
dtype: int64
```

```
#Encoding the categorical data

data = pd.get_dummies(data, columns=['BillNo'], prefix=['BillNo'])
data = pd.get_dummies(data, columns=['Quantity'], prefix=['Quantity'])
```

```
#Handling the duplicates

data.drop_duplicates(inplace=True)
```

(ii)Data Integration

```
#split and load the data set

data=pd.read_csv('/content/Assignment-1_DataN.csv')
data1=pd.read_csv('/content/Assignment-1_DataM.csv')
```

output

```
 <ipython-input-18-c6fb65c16250>:1: DtypeWarning: Columns (0) have mixed types. Specify
dtype option on import or set low_memory=False
```

```
data=pd.read_csv('/content/Assignment-1_DataN.csv')
```

#convert the datasets to data frame

```
 data = pd.DataFrame(data)
data1= pd.DataFrame(data1)
```

#Merging the dataset

```
merged_data = pd.merge(data, data1, on='BillNo')
```

#Printing the merged dataset

```
print(merged_data)
```

output

```
         BillNo  Quantity  Price  CustomerID  \
0        536365         6   2.55     17850.0
1        536365         6   2.55     17850.0
2        536365         6   2.55     17850.0
3        536365         6   2.55     17850.0
4        536365         6   2.55     17850.0
...         ...       ...    ...         ...
14256971  545334         2   1.55     15750.0
14256972  545334         2   1.55     15750.0
14256973  545334         2   1.55     15750.0
14256974  545334         2   1.55     15750.0
14256975  545334         2   1.55     15750.0

                                    Itemname        Date         Country
0         WHITE HANGING HEART T-LIGHT HOLDER  01-12-2010  United Kingdom
1                        WHITE METAL LANTERN  01-12-2010  United Kingdom
2             CREAM CUPID HEARTS COAT HANGER  01-12-2010  United Kingdom
3        KNITTED UNION FLAG HOT WATER BOTTLE  01-12-2010  United Kingdom
4             RED WOOLLY HOTTIE WHITE HEART.  01-12-2010  United Kingdom
...                                      ...         ...             ...
```

```
14256971  PACK OF 6 SANDCASTLE FLAGS ASSORTED  01-03-2011  United Kingdom
14256972               EASTER CRAFT 4 CHICKS  01-03-2011  United Kingdom
14256973          FELTCRAFT BUTTERFLY HEARTS  01-03-2011  United Kingdom
14256974            3 STRIPEY MICE FELTCRAFT  01-03-2011  United Kingdom
14256975        BROWN  PIRATE TREASURE CHEST  01-03-2011        United K

[14256976 rows x 7 columns]
```

(iii)Data Transformation

```
scaler = MinMaxScaler()
merged_data[[ 'Quantity','Price']] =
scaler.fit_transform(merged_data[['Quantity','Price']])
```

#Printing the data after transformation

```
        print(merged_data)
```

output

```
          BillNo  Quantity     Price  CustomerID  \

0            0.0  0.033926  0.000188     17850.0
1            0.0  0.033926  0.000188     17850.0
2            0.0  0.033926  0.000188     17850.0
3            0.0  0.033926  0.000188     17850.0
4            0.0  0.033926  0.000188     17850.0
...          ...       ...       ...         ...
14256971     1.0  0.033874  0.000114     15750.0
14256972     1.0  0.033874  0.000114     15750.0
14256973     1.0  0.033874  0.000114     15750.0
14256974     1.0  0.033874  0.000114     15750.0
14256975     1.0  0.033874  0.000114     15750.0

                                    Itemname        Date         Country
0          WHITE HANGING HEART T-LIGHT HOLDER  01-12-2010  United Kingdom
1                         WHITE METAL LANTERN  01-12-2010  United Kingdom
2              CREAM CUPID HEARTS COAT HANGER  01-12-2010  United Kingdom
3         KNITTED UNION FLAG HOT WATER BOTTLE  01-12-2010  United Kingdom
4              RED WOOLLY HOTTIE WHITE HEART.  01-12-2010  United Kingdom
...                                      ...         ...             ...
14256971  PACK OF 6 SANDCASTLE FLAGS ASSORTED  01-03-2011  United Kingdom
14256972               EASTER CRAFT 4 CHICKS  01-03-2011  United Kingdom
14256973          FELTCRAFT BUTTERFLY HEARTS  01-03-2011  United Kingdom
14256974            3 STRIPEY MICE FELTCRAFT  01-03-2011  United Kingdom
14256975        BROWN  PIRATE TREASURE CHEST  01-03-2011        United K
```

```
[14256976 rows x 7 columns]
```

(iv)Data Reduction

```
pca = PCA(n_components=2)

# Fit and transform your data
reduced_data = pca.fit_transform(data)
```

The code notebook link is given below:

https://colab.research.google.com/drive/1krv0YIVUZQhDk4JfkTmmby6hZ-Xm6ylP?usp=sharing