ARITIFICAL INTELLIGENCE

PHASE-3 SUBMISSION

Market Basket Insights

Data Preprocessing Steps for Market Basket Insights

Data preprocessing is a critical phase in preparing data for analysis, including market basket insights. This document outlines the key steps in data preprocessing without providing the actual code.

List of points for the preprocessing of this dataset:

1. Loading the dataset.
2. Data cleaning, including removing irrelevant columns.
3. Handling missing values, particularly transactions without items.
4. One-hot encoding for categorical data.
5. Splitting the dataset into training and test sets.

Step 1: Loading the Dataset:
Load the dataset into a data analysis tool such as Pandas. Ensure you specify the correct file path and character encoding to read the data correctly.

Step 2: Data Cleaning:
Remove columns that are not relevant to the analysis to reduce the dimensionality of the dataset and improve computational efficiency.

Step 3: Handling Missing Values:
Identify and manage missing data, especially transactions without items. Decide whether to drop, impute, or treat these missing values based on the dataset and analysis goals.

Step 4: One-Hot Encoding:
Convert categorical data into a numerical format suitable for analysis. One-hot encoding is commonly used for this purpose. Each unique category becomes a binary column, indicating the presence or absence of that category.

Step 5: Splitting the Dataset:
Divide the dataset into a training set and a test set. The training set is used for model development, while the test set is reserved for evaluating the model's performance.
These are the fundamental data preprocessing steps for market basket insights. Adapt these steps to your specific dataset and analysis requirements. Data preprocessing is an iterative process, and adjustments may be necessary based on the dataset's characteristics.

Step 1: Load Dataset
Code for the loading the dataset.

```
import pandas as pd
df=pd.read_excel(r'D:\New folder\Assignment-1_Data.xlsx')
print(df.info())
```

Output for the above code:

```
<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
```

Step 2: Data Cleansing:

Code for the cleansing the dataset.

```
import pandas as pd
df = pd.read_excel(r'D:\New folder\Assignment-1_Data.xlsx')
df.dropna(inplace=True)
print(df.info())
```

Output for the above code.

```
<class 'pandas.core.frame.DataFrame'>
Index: 388023 entries, 0 to 522063
Data columns (total 7 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   BillNo      388023 non-null  object
 1   Itemname    388023 non-null  object
 2   Quantity    388023 non-null  int64
 3   Date        388023 non-null  datetime64[ns]
 4   Price       388023 non-null  float64
 5   CustomerID  388023 non-null  float64
 6   Country     388023 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(3)
memory usage: 23.7+ MB
None
```

Step 3: Handling Missing Values:

Code for handling missing values:

```
import pandas as pd
df = pd.read_excel(r'D:\New folder\Assignment-1_Data.xlsx')
missing_values=df.isnull().sum()
print(missing values)
```

Output:

```
BillNo           0
Itemname      1455
Quantity         0
Date             0
Price            0
CustomerID  134041
Country          0
dtype: int64
```

Step 4: One-Hot Encoding:

Code for one-hot encoding.

```
import pandas as pd
df = pd.read_excel(r'D:\New folder\Assignment-1_Data.xlsx')
transaction_item_matrix = pd.get_dummies(df['Itemname']).groupby(df['BillNo']).max()
transaction_item_matrix.fillna(0, inplace=True)
print(transaction_item_matrix.head())
```

Output:

```
        *Boombox Ipod Classic  ...  wrongly sold sets
BillNo                         ...
536365                  False  ...              False
536366                  False  ...              False
536367                  False  ...              False
536368                  False  ...              False
536369                  False  ...              False

[5 rows x 4185 columns]
```

Step 5:

```
import pandas as pd
df = pd.read_excel(r'D:\New folder\Assignment-1_Data.xlsx')
import pandas as pd
data = {
    'BillNo': [536365, 536365, 536365, 536366, 536366, 536367, 536367, 536367],
    'Itemname': [
        'WHITE HANGING HEART T-LIGHT HOLDER',
        'WHITE METAL LANTERN',
        'CREAM CUPID HEARTS COAT HANGER',
        'HAND WARMER UNION JACK',
        'HAND WARMER RED POLKA DOT',
        'ASSORTED COLOUR BIRD ORNAMENT',
        'POPPY\'S PLAYHOUSE BEDROOM',
        'POPPY\'S PLAYHOUSE KITCHEN'
    ]
}
df = pd.DataFrame(data)
dummy_df = pd.get_dummies(df, columns=['Itemname'])
print(dummy_df)
```

```
   BillNo  ...  Itemname_WHITE METAL LANTERN
0  536365  ...                         False
1  536365  ...                          True
2  536365  ...                         False
3  536366  ...                         False
4  536366  ...                         False
5  536367  ...                         False
6  536367  ...                         False
7  536367  ...                         False

[8 rows x 9 columns]
```

TEAM MEMBERS:

1. J. ARUN KUMAR-813821205002
2. C. MURUGANANTHAN-813821205033
3. T. SRIHARIHARAN-813821205049
4. S. VENGADASHAN-813821205055
5. S. VISHWA-813821205059