intrainz

February 27, 2024

[34]:
```
import pandas as pd
```

[35]:
```
import seaborn as sns
```

[36]:
```
file_path = "OnlineRetail_2.xlsx"
```

[37]:
```
data = pd.read_excel(file_path)
```

[38]:
```
print(data.head())
```

```
  InvoiceNo StockCode                          Description  Quantity  \
0    536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER         6
1    536365     71053                  WHITE METAL LANTERN         6
2    536365    84406B       CREAM CUPID HEARTS COAT HANGER         8
3    536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE         6
4    536365    84029E       RED WOOLLY HOTTIE WHITE HEART.         6

          InvoiceDate  UnitPrice  CustomerID         Country
0 2010-12-01 08:26:00       2.55     17850.0  United Kingdom
1 2010-12-01 08:26:00       3.39     17850.0  United Kingdom
2 2010-12-01 08:26:00       2.75     17850.0  United Kingdom
3 2010-12-01 08:26:00       3.39     17850.0  United Kingdom
4 2010-12-01 08:26:00       3.39     17850.0  United Kingdom
```

[39]:
```
print(data.describe())
```

```
            Quantity                    InvoiceDate      UnitPrice  \
count  541909.000000                         541909  541909.000000
mean        9.552250  2011-07-04 13:34:57.156386048       4.611114
min    -80995.000000            2010-12-01 08:26:00  -11062.060000
25%         1.000000            2011-03-28 11:34:00       1.250000
50%         3.000000            2011-07-19 17:17:00       2.080000
75%        10.000000            2011-10-19 11:27:00       4.130000
max     80995.000000            2011-12-09 12:50:00   38970.000000
std       218.081158                            NaN      96.759853

          CustomerID
count  406829.000000
```

```
mean    15287.690570
min     12346.000000
25%     13953.000000
50%     15152.000000
75%     16791.000000
max     18287.000000
std      1713.600303
```

[40]:
```python
def find_popular_items_globally(data):
    popular_items = data.groupby('Description')['Quantity'].sum().
 ↪sort_values(ascending=False)
    return popular_items
```

[41]:
```python
def find_popular_items_country_wise(data, country):
    popular_items = data[data['Country'] == country].
 ↪groupby('Description')['Quantity'].sum().sort_values(ascending=False)
    return popular_items
```

[42]:
```python
def find_popular_items_month_wise(data, month):
    data['InvoiceDate'] = pd.to_datetime(data['InvoiceDate'])
    data['Month'] = data['InvoiceDate'].dt.month
    popular_items = data[data['Month'] == month].
 ↪groupby('Description')['Quantity'].sum().sort_values(ascending=False)
    return popular_items
```

[43]:
```python
def analyze_and_print_recommendation(data, item):
    if item in data['Description'].values:
        recommendation = data[data['Description'] == item]
        print("Recommendation for", item, ":\n", recommendation)
    else:
        print("Item not found in the dataset.")
```

[56]:
```python
def predict(data, customer_id):
    try:
        customer_data = data[data['CustomerID'] == customer_id]
        if len(customer_data) > 0:
            popular_items = customer_data.groupby('Description')['Quantity'].
 ↪sum().sort_values(ascending=False)
            if len(popular_items) > 0:
                return popular_items.index[0]  # Return the most popular item␣
 ↪for the customer
            else:
                return "No purchases found for the customer."
        else:
            return "No data found for the customer ID."
    except KeyError:
        print("Error: 'CustomerID' column not found in the dataset.")
```

```
    except Exception as e:
        print("An error occurred:", e)
```

[45]:
```
popular_items_globally = find_popular_items_globally(data)
print("Popular items globally:\n", popular_items_globally.head(10))
```

```
Popular items globally:
 Description
WORLD WAR 2 GLIDERS ASSTD DESIGNS     53847
JUMBO BAG RED RETROSPOT               47363
ASSORTED COLOUR BIRD ORNAMENT         36381
POPCORN HOLDER                        36334
PACK OF 72 RETROSPOT CAKE CASES       36039
WHITE HANGING HEART T-LIGHT HOLDER    35317
RABBIT NIGHT LIGHT                    30680
MINI PAINT SET VINTAGE                26437
PACK OF 12 LONDON TISSUES             26315
PACK OF 60 PINK PAISLEY CAKE CASES    24753
Name: Quantity, dtype: int64
```

[46]:
```
popular_items_country_wise = find_popular_items_country_wise(data, 'United␣
↪Kingdom')
print("Popular items in United Kingdom:\n", popular_items_country_wise.head(10))
```

```
Popular items in United Kingdom:
 Description
WORLD WAR 2 GLIDERS ASSTD DESIGNS     48326
JUMBO BAG RED RETROSPOT               43167
POPCORN HOLDER                        34365
ASSORTED COLOUR BIRD ORNAMENT         33679
WHITE HANGING HEART T-LIGHT HOLDER    33193
PACK OF 12 LONDON TISSUES             25307
PACK OF 72 RETROSPOT CAKE CASES       24702
VICTORIAN GLASS HANGING T-LIGHT       23242
BROCADE RING PURSE                    22801
ASSORTED COLOURS SILK FAN             20322
Name: Quantity, dtype: int64
```

[47]:
```
def find_popular_items_month_wise(data, month):
    try:
        # Convert 'Invoice Date' to datetime if not already done
        if 'InvoiceDate' not in data:
            data['InvoiceDate'] = pd.to_datetime(data['InvoiceDate'])

        # Extract month
        data['Month'] = data['InvoiceDate'].dt.month
```

```
        # Filter data for the specified month
        data_month = data[data['Month'] == month]

        # Group by 'Description' and sum 'Quantity' for popular items
        popular_items_month_wise = data_month.
 ↪groupby('Description')['Quantity'].sum().sort_values(ascending=False)
        return popular_items_month_wise
    except KeyError:
        print("Error: 'InvoiceDate' column not found in the dataset.")
    except Exception as e:
        print("An error occurred:", e)
```

[48]:
```
popular_items_month_wise = find_popular_items_month_wise(data, 1)  # Assuming␣
 ↪month is January
if popular_items_month_wise is not None:
    print("Popular items in January:\n", popular_items_month_wise.head(10))
```

```
Popular items in January:
 Description
WHITE HANGING HEART T-LIGHT HOLDER    5530
PACK OF 72 RETROSPOT CAKE CASES       3409
FAIRY CAKE FLANNEL ASSORTED COLOUR    3246
BROCADE RING PURSE                    3181
HEART OF WICKER SMALL                 2919
JUMBO BAG RED RETROSPOT               2639
GIN + TONIC DIET METAL SIGN           2410
CHINESE DRAGON PAPER LANTERNS         2128
MINI PAINT SET VINTAGE                1913
SMALL POPCORN HOLDER                  1809
Name: Quantity, dtype: int64
```

[65]:
```
analyze_and_print_recommendation(data, "NATURAL SLATE HEART CHALKBOARD")
```

```
Recommendation for NATURAL SLATE HEART CHALKBOARD :
        InvoiceNo StockCode                      Description  Quantity  \
159        536384     22457  NATURAL SLATE HEART CHALKBOARD        12
1357       536538     22457  NATURAL SLATE HEART CHALKBOARD         2
1582       536544     22457  NATURAL SLATE HEART CHALKBOARD         6
2254       536570     22457  NATURAL SLATE HEART CHALKBOARD        12
2412       536590     22457  NATURAL SLATE HEART CHALKBOARD         6
…             …        …                                …        …
538931     581424     22457  NATURAL SLATE HEART CHALKBOARD         6
539598     581439     22457  NATURAL SLATE HEART CHALKBOARD         8
540368     581477     22457  NATURAL SLATE HEART CHALKBOARD         6
540922     581492     22457  NATURAL SLATE HEART CHALKBOARD         1
541427     581498     22457  NATURAL SLATE HEART CHALKBOARD         2
```

```
                InvoiceDate  UnitPrice  CustomerID         Country  Month
159    2010-12-01 09:53:00       2.95     18074.0  United Kingdom     12
1357   2010-12-01 13:54:00       2.95     14594.0  United Kingdom     12
1582   2010-12-01 14:32:00       5.91         NaN  United Kingdom     12
2254   2010-12-01 15:35:00       2.95     14496.0  United Kingdom     12
2412   2010-12-01 16:52:00       2.95     13065.0  United Kingdom     12
…                      …          …           …              …    …
538931 2011-12-08 15:29:00       2.95     13890.0  United Kingdom     12
539598 2011-12-08 16:30:00       5.79         NaN  United Kingdom     12
540368 2011-12-09 08:59:00       2.95     13426.0  United Kingdom     12
540922 2011-12-09 10:03:00       5.79         NaN  United Kingdom     12
541427 2011-12-09 10:26:00       5.79         NaN  United Kingdom     12

[1280 rows x 9 columns]
```

[71]:
```
predicted_item = predict(data,16729)
print("Predicted item for customer 16729:", predicted_item)
```

```
Predicted item for customer 16729: TRAVEL CARD WALLET KEEP CALM
```

[ ]:

[ ]:

[ ]: