

ONLINE RETAIL PROJECT

Input code for the system:

```
import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.metrics.pairwise import cosine_similarity

from sklearn.preprocessing import StandardScaler

import numpy as np

data = pd.read_csv('/users/lakshaypawar/downloads/onlineretail .csv')

print(data.head())

# Data Description

print(data.describe())

print(data.info())

missing_values = data.isnull().sum()

data_cleaned = data.dropna(subset=['CustomerID', 'Description'])

pivot_table = data_cleaned.pivot_table(index='CustomerID', columns='Description',
values='Quantity', aggfunc='sum', fill_value=0)

scaler = StandardScaler()

pivot_scaled = scaler.fit_transform(pivot_table)

product_similarity = cosine_similarity(pivot_scaled)

product_similarity_df = pd.DataFrame(product_similarity, index=pivot_table.index,
columns=pivot_table.index)

popular_items_global =

data_cleaned.groupby('Description')['Quantity'].sum().sort_values(ascending=False).head(10)

print("Top 10 Popular Items Globally:\n", popular_items_global)
```

```
popular_items_country = data_cleaned.groupby(['Country',  
'Description'])['Quantity'].sum().sort_values(ascending=False).groupby(level=0).head(1)  
  
print("Popular Items by Country:\n", popular_items_country)  
  
  
data_cleaned['InvoiceDate'] = pd.to_datetime(data_cleaned['InvoiceDate'])  
  
data_cleaned['Month'] = data_cleaned['InvoiceDate'].dt.month  
  
popular_items_month = data_cleaned.groupby(['Month',  
'Description'])['Quantity'].sum().sort_values(ascending=False).groupby(level=0).head(1)  
  
print("Popular Items by Month:\n", popular_items_month)  
  
print(pivot_table.index)  
  
print(pivot_table.index.dtype)  
  
def recommend_products(customer_id, pivot_table, product_similarity_df, n=5):  
  
    customer_purchases = pivot_table.loc[customer_id]  
  
  
    similar_customers = product_similarity_df[customer_id].sort_values(ascending=False)  
  
  
    similar_customer_ids = similar_customers.index  
  
  
    similar_customer_purchases = pivot_table.loc[similar_customer_ids].sum(axis=0)  
  
  
    products_to_recommend = similar_customer_purchases[customer_purchases ==  
0].sort_values(ascending=False)  
  
  
    recommendations = products_to_recommend.head(n)  
  
  
    return recommendations  
  
customer_id = 17850 # Replace with an actual Customer ID from the dataset  
  
recommendations = recommend_products(customer_id, pivot_table, product_similarity_df)  
  
print(f"Recommended Products for Customer {customer_id}:\n", recommendations)  
  
sns.barplot(x=popular_items_global.values, y=popular_items_global.index)  
  
plt.title("Top 10 Globally Popular Items")  
  
plt.xlabel("Quantity Sold")
```

```

plt.ylabel("Product Description")

plt.show()

```

result for the input code:

The screenshot shows a Jupyter Notebook interface with the title "jupyter onlineretail Last Checkpoint: 3 days ago". The menu bar includes File, Edit, View, Run, Kernel, Settings, Help, and a Trusted button. The toolbar includes icons for file operations like New, Open, Save, and Run, along with JupyterLab and Python 3 (ipykernel) buttons.

```

[15]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.preprocessing import StandardScaler
import numpy as np

* [17]: data = pd.read_csv('/users/lakshaypawar/downloads/onlineretail .csv')

print(data.head())

```

The output displays the first 5 rows of the dataset:

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

InvoiceDate UnitPrice CustomerID Country

0	12/1/2010 8:26	2.55	17850.0 United Kingdom
1	12/1/2010 8:26	3.39	17850.0 United Kingdom
2	12/1/2010 8:26	2.75	17850.0 United Kingdom
3	12/1/2010 8:26	3.39	17850.0 United Kingdom
4	12/1/2010 8:26	3.39	17850.0 United Kingdom

The screenshot shows a Jupyter Notebook interface with the title "jupyter onlineretail Last Checkpoint: 3 days ago". The menu bar includes File, Edit, View, Run, Kernel, Settings, Help, and a Trusted button. The toolbar includes icons for file operations like New, Open, Save, and Run, along with JupyterLab and Python 3 (ipykernel) buttons.

```

[19]: # Data Description
print(data.describe())
print(data.info())

```

The output displays descriptive statistics and information about the DataFrame:

	Quantity	UnitPrice	CustomerID
count	541909.000000	541909.000000	406829.000000
mean	9.552250	4.611114	15287.690570
std	218.081158	96.759853	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 # Column Non-Null Count Dtype
 --- -- -- -- --
 0 InvoiceNo 541909 non-null object
 1 StockCode 541909 non-null object
 2 Description 540455 non-null object
 3 Quantity 541909 non-null int64
 4 InvoiceDate 541909 non-null object
 5 UnitPrice 541909 non-null float64
 6 CustomerID 406829 non-null float64
 7 Country 541909 non-null object
 dtypes: float64(2), int64(1), object(5)
 memory usage: 33.1+ MB
 None

```

* [23]: missing_values = data.isnull().sum()

data_cleaned = data.dropna(subset=['CustomerID', 'Description'])

```

```
[27]: pivot_table = data_cleaned.pivot_table(index='CustomerID', columns='Description', values='Quantity', aggfunc='sum', fill_value=0)

scaler = StandardScaler()
pivot_scaled = scaler.fit_transform(pivot_table)

[33]: product_similarity = cosine_similarity(pivot_scaled)

product_similarity_df = pd.DataFrame(product_similarity, index=pivot_table.index, columns=pivot_table.index)
```

jupyter onlineretail Last Checkpoint: 3 days ago

File Edit View Run Kernel Settings Help Trusted JupyterLab Python 3 (ipykernel)

```
[35]: popular_items_global = data_cleaned.groupby('Description')['Quantity'].sum().sort_values(ascending=False).head(10)
print("Top 10 Popular Items Globally:\n", popular_items_global)

popular_items_country = data_cleaned.groupby(['Country', 'Description'])['Quantity'].sum().sort_values(ascending=False).groupby(level=0).head(1)
print("Popular Items by Country:\n", popular_items_country)

data_cleaned['InvoiceDate'] = pd.to_datetime(data_cleaned['InvoiceDate'])
data_cleaned['Month'] = data_cleaned['InvoiceDate'].dt.month
popular_items_month = data_cleaned.groupby(['Month', 'Description'])['Quantity'].sum().sort_values(ascending=False).groupby(level=0).head(1)
print("Popular Items by Month:\n", popular_items_month)
```

Top 10 Popular Items Globally:

Description	Quantity
WORLD WAR 2 GLIDERS ASSTD DESIGNS	53215
JUMBO BAG RED RETROSPOT	45666
ASSORTED COLOUR BIRD ORNAMENT	35314
WHITE HANGING HEART T-LIGHT HOLDER	34147
PACK OF 72 RETROSPOT CAKE CASES	33409
POPCORN HOLDER	30504
RABBIT NIGHT LIGHT	27094
MINI PAINT SET VINTAGE	25880
PACK OF 12 LONDON TISSUES	25321
PACK OF 60 PINK PAISLEY CAKE CASES	24163

Name: Quantity, dtype: int64

Popular Items by Country:

Country	Description	Quantity
United Kingdom	WORLD WAR 2 GLIDERS ASSTD DESIGNS	47982
Netherlands	RABBIT NIGHT LIGHT	4801
France	RABBIT NIGHT LIGHT	3999
Japan	RABBIT NIGHT LIGHT	3401
Sweden	MINI PAINT SET VINTAGE	2916
Australia	MINI PAINT SET VINTAGE	2916
EIRE	PACK OF 72 RETROSPOT CAKE CASES	1632
Germany	ROUND SNACK BOXES SET OF4 WOODLAND	1218
Spain	CHILDRENS CUTLERY POLKADOT PINK	729
Switzerland	PLASTERS IN TIN WOODLAND ANIMALS	636
Norway	SMALL FOLDING SCISSOR(POINTED EDGE)	576
Canada	RETRO COFFEE MUG ASSORTED	524

jupyter onlineretail Last Checkpoint: 3 days ago

File Edit View Run Kernel Settings Help Trusted JupyterLab Python 3 (ipykernel)

```
Finland CHILDRENS CUTLERY POLKADOT PINK 480
Channel Islands RAIN PONCHO 407
Singapore CHRISTMAS TREE PAINTED ZINC 384
Cyprus HEART DECORATION PAINTED ZINC 384
Denmark RED HARMONICA IN BOX 288
Austria SET 12 KIDS COLOUR CHALK STICKS 288
Portugal POLKADOT PEN 240
Italy FEATHER PEN,HOT PINK 240
Iceland ICE CREAM SUNDAE LIP GLOSS 240
Poland STRAWBERRY CERAMIC TRINKET BOX 144
Israel WOODLAND CHARLOTTE BAG 120
Bahrain ICE CREAM SUNDAE LIP GLOSS 96
USA SET 12 COLOURING PENCILS DOILY 88
Malta GRAND CHOCOLATECANDLE 75
United Arab Emirates ASSORTED CHEESE FRIDGE MAGNETS 72
Czech Republic WOODEN STAR CHRISTMAS SCANDINAVIAN 72
Greece 4 LAVENDER BOTANICAL DINNER CANDLES 48
Lithuania FELTCRAFT DOLL ROSIE 48
Unspecified PACK OF 6 LARGE FRUIT STRAWS 36
Brazil DOLLY GIRL LUNCH BOX 24
European Community WHITE ROCKING HORSE HAND PAINTED 24
Lebanon ASSTD FRUIT+FLOWERS FRIDGE MAGNETS 24
RSA 4 TRADITIONAL SPINNING TOPS 12
Saudi Arabia ASSORTED BOTTLE TOP MAGNETS 12
Name: Quantity, dtype: int64
Popular Items by Month:
```

Month	Description	Quantity
11	ASSTD DESIGN 3D PAPER STICKERS	12551
4	WORLD WAR 2 GLIDERS ASSTD DESIGNS	10128
10	WORLD WAR 2 GLIDERS ASSTD DESIGNS	8174
5	SMALL POPCORN HOLDER	6737
12	WORLD WAR 2 GLIDERS ASSTD DESIGNS	6502
8	ASSORTED COLOUR BIRD ORNAMENT	6424
1	WHITE HANGING HEART T-LIGHT HOLDER	5456
3	JUMBO BAG RED RETROSPOT	4923
7	GIRLS ALPHABET IRON ON PATCHES	4896
9	JUMBO BAG RED RETROSPOT	4043
2	EMPIRE DESIGN ROSETTE	3986
6	JUMBO BAG RED RETROSPOT	3523

Name: Quantity, dtype: int64

jupyter onlineretail Last Checkpoint: 3 days ago

File Edit View Run Kernel Settings Help Trusted JupyterLab Python 3 (ipykernel)

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
data_cleaned['InvoiceDate'] = pd.to_datetime(data_cleaned['InvoiceDate'])
/var/folders/ss/dlrb185dc59scfg63zbg27r0000gn/T/ipykernel_11425/361076366.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
data_cleaned['Month'] = data_cleaned['InvoiceDate'].dt.month
```

```
[41]: print(pivot_table.index)
print(pivot_table.index.dtype)
```

```
Index([12346.0, 12347.0, 12348.0, 12349.0, 12350.0, 12352.0, 12353.0, 12354.0,
       12355.0, 12356.0,
       ...
       18273.0, 18274.0, 18276.0, 18277.0, 18278.0, 18280.0, 18281.0, 18282.0,
       18283.0, 18287.0],
      dtype='float64', name='CustomerID', length=4372)
float64
```

```
* [45]: 
```

```
* [61]: def recommend_products(customer_id, pivot_table, product_similarity_df, n=5):
    customer_purchases = pivot_table.loc[customer_id]

    similar_customers = product_similarity_df[customer_id].sort_values(ascending=False)

    similar_customer_ids = similar_customers.index

    similar_customer_purchases = pivot_table.loc[similar_customer_ids].sum(axis=0)

    products_to_recommend = similar_customer_purchases[customer_purchases == 0].sort_values(ascending=False)

    recommendations = products_to_recommend.head(n)

    return recommendations
```

```
* [63]: customer_id = 17850 # Replace with an actual Customer ID from the dataset
recommendations = recommend_products(customer_id, pivot_table, product_similarity_df)
print(f"Recommended Products for Customer {customer_id}:\n", recommendations)
```

```
Recommended Products for Customer 17850:
Description
WORLD WAR 2 GLIDERS ASSTD DESIGNS      53215
JUMBO BAG RED RETROSPOT                 45066
ASSORTED COLOUR BIRD ORNAMENT           35314
PACK OF 72 RETROSPOT CAKE CASES        33409
POPCORN HOLDER                           30504
dtype: int64
```

