

import all libraries

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import train_test_split
from sklearn.metrics import mean_squared_error
from math import sqrt

from sklearn.preprocessing import LabelEncoder
Country = LabelEncoder().fit_transform(
    from sklearn.metrics import accuracy_score, classification_report

In [3]: df=pd.read_csv("C:\Users\User\Desktop\OnlineRetail.csv")

In [4]: df

Out[4]:
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12-1-2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12-1-2010 8:26	3.39	17850.0	United Kingdom
2	536365	840068	CREAM CUPID HEARTS COAT HANGER	8	12-1-2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12-1-2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12-1-2010 8:26	3.39	17850.0	United Kingdom
541904	581587	22013	PACK OF 20 SPACEBOY NAPKINS	12	12-9-2011 12:50	0.85	12680.0	France
541905	581587	22999	CHILDRENS APRON DOLLY GIRL	6	12-9-2011 12:50	2.10	12680.0	France
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	12-9-2011 12:50	4.15	12680.0	France
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	12-9-2011 12:50	4.15	12680.0	France
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	12-9-2011 12:50	4.95	12680.0	France

541500 rows x 8 columns

```
In [5]: df.head()
```

```
Out[5]:
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12-1-2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12-1-2010 8:26	3.39	17850.0	United Kingdom
2	536365	840068	CREAM CUPID HEARTS COAT HANGER	8	12-1-2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12-1-2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12-1-2010 8:26	3.39	17850.0	United Kingdom

```
In [6]: df.info()
```

```
<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 486829 non-null float64
7 Country 541909 non-null object
dtypes: float64(2), int64(1), object(5)
memory usage: 33.1+ MB
```

```
In [7]: df.shape
```

```
Out[7]: (541909, 8)
```

description

```
In [8]: df.describe()
```

```
Out[8]:
```

	Quantity	UnitPrice	CustomerID
count	541909.000000	541909.000000	486829.000000
mean	5.922220	4.611114	15387.590700
std	218.081156	96.759893	1713.697000
min	1.000000	1.120500	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16781.000000
max	80095.000000	38970.000000	18287.000000

```
In [9]: df[['StockCode','Description','InvoiceDate','Country']].dropna()
```

```
Out[9]:
```

	StockCode	Description	InvoiceDate	Country
0	85123A	WHITE HANGING HEART T-LIGHT HOLDER	12-1-2010 8:26	United Kingdom
1	71053	WHITE METAL LANTERN	12-1-2010 8:26	United Kingdom
2	840068	CREAM CUPID HEARTS COAT HANGER	12-1-2010 8:26	United Kingdom
3	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	12-1-2010 8:26	United Kingdom
4	84029E	RED WOOLLY HOTTIE WHITE HEART.	12-1-2010 8:26	United Kingdom
541904	22613	PACK OF 20 SPACEBOY NAPKINS	12-9-2011 12:50	France
541905	22999	CHILDRENS APRON DOLLY GIRL	12-9-2011 12:50	France
541906	23254	CHILDRENS CUTLERY DOLLY GIRL	12-9-2011 12:50	France
541907	23255	CHILDRENS CUTLERY CIRCUS PARADE	12-9-2011 12:50	France
541908	22138	BAKING SET 9 PIECE RETROSPOT	12-9-2011 12:50	France

540455 rows x 4 columns

most popular item by globally

```
In [10]: item_counts = df['Description'].value_counts()
item_counts

Out[10]:
WHITE HANGING HEART T-LIGHT HOLDER    2360
REGENCY CAKESTAND 3 TIER              2289
JUMBO BAG RED RETROSPOT               2159
PARTY BUNTING                        1727
LUNCH BAG RED RETROSPOT               1638
...
stock credited wrongly                 1
MACE ANTOINETT TRIMMET BOX GOLD      1
label mix up                          1
DOWNHILL CERAMIC IVORY               1
SET OF 3 PINK FLYING DUCKS           1
Name: Description, Length: 4231, dtype: int64

In [11]: most_popular_item = item_counts.idxmax()
most_popular_item

Out[11]: 'WHITE HANGING HEART T-LIGHT HOLDER'
```

most popular item by country wise

```
In [12]: ctry=cvs['Country'].unique()
ctry

Out[12]: array(['United Kingdom', 'France', 'Australia', 'Netherlands', 'Germany',
              'Netherlands', 'Ireland', 'Switzerland', 'Spain', 'Poland', 'Portugal',
              'Italy', 'Belgium', 'Lithuania', 'Japan', 'Iceland',
              'Channel Islands', 'Denmark', 'Cyprus', 'Sweden', 'Austria',
              'Israel', 'Finland', 'Bahrain', 'Greece', 'Hong Kong', 'Singapore',
              'Lebanon', 'United Arab Emirates', 'Saudi Arabia',
              'Czech Republic', 'Canada', 'Unspecified', 'Brazil', 'USA',
              'European Community', 'Malta', 'RSA'], dtype=object)

In [13]: desc=cvs['Description'].unique()
desc

Out[13]: array(['WHITE HANGING HEART T-LIGHT HOLDER', 'WHITE METAL LANTERN',
              'CREAM CUPID HEARTS COAT HANGER', ..., 'Jest',
              'CREAM HANGING HEART T-LIGHT HOLDER',
              'PAPER CHAIN KIT 30'S CHRISTMAS', 'dtype=object)

In [14]: #Group data by country
grouped = df.groupby('Country')

# Define a popularity metric (e.g., sales or ratings)
# You can replace 'Sales' with your actual metric column
popularity_metric = 'StockCode'

popularity_items = []

# Iterate through each group (country)
for country, group in grouped:
    # Sort items within each country by popularity metric
    sorted_group = group.sort_values(by=popularity_metric, ascending=False)

    # Select the most popular item (the first one after sorting)
    popular_item = sorted_group.iloc[0]

    # Append the popular item to the list
    popularity_items.append(popular_item)

# Display the popular items in each country
for item in popularity_items:
    print(f'Country: {item["Country"]}, Popular Item: {item["Description"]}')

Country: Australia, Popular Item: POSTAGE
Country: Austria, Popular Item: POSTAGE
Country: Bahrain, Popular Item: S/4 PINK FLORER CANDLES IN BOWL
Country: Belgium, Popular Item: POSTAGE
Country: Brazil, Popular Item: 3x4.1cm HEART FLOWERS HOOK
Country: Canada, Popular Item: POSTAGE
Country: Channel Islands, Popular Item: Manual
Country: Cyprus, Popular Item: POSTAGE
Country: Czech Republic, Popular Item: POSTAGE
Country: Denmark, Popular Item: POSTAGE
Country: EIRE, Popular Item: Manual
Country: European Community, Popular Item: POSTAGE
Country: Finland, Popular Item: POSTAGE
Country: France, Popular Item: POSTAGE
Country: Germany, Popular Item: POSTAGE
Country: Greece, Popular Item: POSTAGE
Country: Hong Kong, Popular Item: POSTAGE
Country: Iceland, Popular Item: SET/3 DECORATIVE STACKING TINS
Country: Israel, Popular Item: NECKLACE+BRACELET SET BLUE HIBISCUS
Country: Italy, Popular Item: POSTAGE
Country: Japan, Popular Item: JUMBO BAG RED RETROSPOT
Country: Lebanon, Popular Item: LADIES & GENTLEMEN METAL SIGN
Country: Lithuania, Popular Item: CREAM FILT EASTER EGG BASKET
Country: Malta, Popular Item: POSTAGE
Country: Netherlands, Popular Item: POSTAGE
Country: Norway, Popular Item: POSTAGE
Country: Poland, Popular Item: POSTAGE
Country: Portugal, Popular Item: POSTAGE
Country: RSA, Popular Item: Manual
Country: Saudi Arabia, Popular Item: HOMEWARE JAM SCENTED CANDLES
Country: Singapore, Popular Item: Manual
Country: Spain, Popular Item: POSTAGE
Country: Sweden, Popular Item: POSTAGE
Country: Switzerland, Popular Item: POSTAGE
Country: USA, Popular Item: JUMBO BAG RED RETROSPOT
Country: United Arab Emirates, Popular Item: POSTAGE
Country: United Kingdom, Popular Item: Manual
Country: Unspecified, Popular Item: SET OF 6 3D KIT CARDS FOR KIDS
```

popular item with month number

```
In [16]: # Convert the 'date' column to datetime
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])

# Extract the month from the 'date' column
df['month'] = df['InvoiceDate'].dt.month

# Group by 'month' and 'item', then count occurrences
popularity_items_by_month = df.groupby(['month', 'Description']).size().reset_index(name='count')

# Find the most popular item in each month
most_popular_items = popularity_items_by_month.apply(lambda x: x['count'].max())

print(popularity_items_by_month)
```

	month	Description	count
1	2446	1 WHITE HANGING HEART T-LIGHT HOLDER	195
2	4422	2 SET OF 3 CAKE TINS PANTRY DESIGN	157
3	8744	3 REGENCY CAKESTAND 3 TIER	228
4	8932	4 PARTY BUNTING	195
5	11389	5 PARTY BUNTING	253
6	13847	6 PARTY BUNTING	211
7	16611	7 PARTY BUNTING	187
8	18843	8 JUMBO BAG RED RETROSPOT	192
9	21583	9 JUMBO BAG RED RETROSPOT	228
10	24749	10 PAPER CHAIN KIT 30'S CHRISTMAS	237
11	27584	11 RABBIT NIGHT LIGHT	523
12	32221	12 WHITE HANGING HEART T-LIGHT HOLDER	304

most popular item by month name

```
In [16]: # Parse the date column and extract month and year
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])
df['month'] = df['InvoiceDate'].dt.strftime('%b-%y')

# Group by month and count occurrences of each item
monthly_counts = df.groupby('Month')['Description'].value_counts().reset_index(name='Count')

# Find the most popular item in each month
most_popular_items = monthly_counts.groupby('Month').apply(lambda x: x.nlargest(1, 'Count')).reset_index(drop=True)

# Map month names to month-year values
most_popular_items['Month'] = pd.to_datetime(most_popular_items['Month'] + '-01').dt.strftime('%b %Y')

# Display the results
print(most_popular_items)
```

	Month	Description	Count
0	December 2010	WHITE HANGING HEART T-LIGHT HOLDER	241
1	January 2011	WHITE HANGING HEART T-LIGHT HOLDER	185
2	February 2011	SET OF 3 CAKE TINS PANTRY DESIGN	157
3	March 2011	REGENCY CAKESTAND 3 TIER	228
4	April 2011	PARTY BUNTING	195
5	May 2011	PARTY BUNTING	253
6	June 2011	PARTY BUNTING	211
7	July 2011	PARTY BUNTING	187
8	August 2011	JUMBO BAG RED RETROSPOT	192
9	September 2011	JUMBO BAG RED RETROSPOT	228
10	October 2011	PAPER CHAIN KIT 30'S CHRISTMAS	237
11	November 2011	RABBIT NIGHT LIGHT	523
12	December 2011	RABBIT NIGHT LIGHT	159

PIVOTTABLE

```
In [18]: # Create a pivot table
pivot_table = pd.pivot_table(df, values='Quantity', index='Country', columns='InvoiceDate', aggfunc='sum')

# Display the pivot table
print(pivot_table)
```

InvoiceDate	2010-12-01 08:26:00	2010-12-01 08:26:00 \
Country		
Australia	NaN	NaN
Austria	NaN	NaN
Bahrain	NaN	NaN
Belgium	NaN	NaN
Brazil	NaN	NaN
Canada	NaN	NaN
Channel Islands	NaN	NaN
Cyprus	NaN	NaN
Czech Republic	NaN	NaN
Denmark	NaN	NaN
EIRE	NaN	NaN
European Community	NaN	NaN
Finland	NaN	NaN
France	NaN	NaN
Germany	NaN	NaN
Greece	NaN	NaN
Hong Kong	NaN	NaN
Iceland	NaN	NaN
Ireland	NaN	NaN
Israel	NaN	NaN
Italy	NaN	NaN
Japan	NaN	NaN
Lebanon	NaN	NaN
Lithuania	NaN	NaN
Malta	NaN	NaN
Netherlands	NaN	NaN
Norway	NaN	NaN
Poland	NaN	NaN
Portugal	NaN	NaN
RSA	NaN	NaN
Saudi Arabia	NaN	NaN
Singapore	NaN	NaN
Spain	NaN	NaN
Sweden	NaN	NaN
Switzerland	NaN	NaN
USA	NaN	NaN
United Arab Emirates	NaN	NaN
United Kingdom	48.0	12.0
Unspecified	NaN	NaN

InvoiceDate	2010-12-01 08:34:00	2010-12-01 08:35:00 \
Country		
Australia	NaN	NaN
Austria	NaN	NaN
Bahrain	NaN	NaN
Belgium	NaN	NaN
Brazil	NaN	NaN
Canada	NaN	NaN
Channel Islands	NaN	NaN
Cyprus	NaN	NaN
Czech Republic	NaN	NaN
Denmark	NaN	NaN
EIRE	NaN	NaN
European Community	NaN	NaN
Finland	NaN	NaN
France	448.0	NaN
Germany	NaN	NaN
Greece	NaN	NaN
Hong Kong	NaN	NaN
Iceland	NaN	NaN
Ireland	NaN	NaN
Israel	NaN	NaN
Italy	NaN	NaN
Japan	NaN	NaN
Lebanon	NaN	NaN
Lithuania	NaN	NaN
Malta	NaN	NaN
Netherlands	NaN	NaN
Norway	NaN	NaN
Poland	NaN	NaN
Portugal	NaN	NaN
RSA	NaN	NaN
Saudi Arabia	NaN	NaN
Singapore	NaN	NaN
Spain	NaN	NaN
Sweden	NaN	NaN
Switzerland	NaN	NaN
USA	NaN	NaN
United Arab Emirates	NaN	NaN
United Kingdom	90.0	3.0
Unspecified	NaN	NaN

InvoiceDate	2010-12-01 08:45:00	2010-12-01 09:00:00 \
Country		
Australia	NaN	NaN
Austria	NaN	NaN
Bahrain	NaN	NaN
Belgium	NaN	NaN
Brazil	NaN	NaN
Canada	NaN	NaN
Channel Islands	NaN	NaN
Cyprus	NaN	NaN
Czech Republic	NaN	NaN
Denmark	NaN	NaN
EIRE	NaN	NaN
European Community	NaN	NaN
Finland	NaN	NaN
France	448.0	NaN
Germany	NaN	NaN
Greece	NaN	NaN
Hong Kong	NaN	NaN
Iceland	NaN	NaN
Ireland	NaN	NaN
Israel	NaN	NaN
Italy	NaN	NaN
Japan	NaN	NaN
Lebanon	NaN	NaN
Lithuania	NaN	NaN
Malta	NaN	NaN
Netherlands	NaN	NaN
Norway	NaN	NaN
Poland	NaN	NaN
Portugal	NaN	NaN
RSA	NaN	NaN
Saudi Arabia	NaN	NaN
Singapore	NaN	NaN
Spain	NaN	NaN
Sweden	NaN	NaN
Switzerland	NaN	NaN
USA	NaN	NaN
United Arab Emirates	NaN	NaN
United Kingdom	32.0	260.0
Unspecified	NaN	NaN

InvoiceDate	2010-12-01 09:01:00	2010-12-01 09:02:00 \
Country		
Australia	NaN	NaN
Austria	NaN	NaN
Bahrain	NaN	NaN
Belgium	NaN	NaN
Brazil	NaN	NaN
Canada	NaN	NaN
Channel Islands	NaN	NaN
Cyprus	NaN	NaN
Czech Republic	NaN	NaN
Denmark	NaN	NaN
EIRE	NaN	NaN
European Community	NaN	NaN
Finland	NaN	NaN
France	NaN	NaN
Germany	NaN	NaN
Greece	NaN	NaN
Hong Kong	NaN	NaN
Iceland	NaN	NaN
Ireland	NaN	NaN
Israel	NaN	NaN
Italy	NaN	NaN
Japan	NaN	NaN
Lebanon	NaN	NaN
Lithuania	NaN	NaN
Malta	NaN	NaN
Netherlands	NaN	NaN
Norway	NaN	NaN
Poland	NaN	NaN
Portugal	NaN	NaN
RSA	NaN	NaN
Saudi Arabia	NaN	NaN
Singapore	NaN	NaN
Spain	NaN	NaN
Sweden	NaN	NaN
Switzerland	NaN	NaN
USA	NaN	NaN
United Arab Emirates	NaN	NaN
United Kingdom	12.0	80.0
Unspecified	NaN	NaN

InvoiceDate	2010-12-01 09:09:00	2010-12-01 09:32:00 \
Country		
Australia	NaN	NaN
Austria	NaN	NaN
Bahrain	NaN	NaN
Belgium	NaN	NaN
Brazil	NaN	NaN
Canada	NaN	NaN
Channel Islands	NaN	NaN
Cyprus	NaN	NaN
Czech Republic	NaN	NaN
Denmark	NaN	NaN
EIRE	NaN	NaN
European Community	NaN	NaN
Finland	NaN	NaN
France	448.0	NaN
Germany	NaN	NaN
Greece	NaN	NaN
Hong Kong	NaN	NaN
Iceland	NaN	NaN
Ireland	NaN	NaN
Israel	NaN	NaN
Italy	NaN	NaN
Japan	NaN	NaN
Lebanon	NaN	NaN
Lithuania	NaN	NaN
Malta	NaN	NaN
Netherlands	NaN	NaN
Norway	NaN	NaN
Poland	NaN	NaN
Portugal	NaN	NaN
RSA	NaN	NaN
Saudi Arabia	NaN	NaN
Singapore	NaN	NaN
Spain	NaN	NaN
Sweden	NaN	NaN
Switzerland	NaN	NaN
USA	NaN	NaN
United Arab Emirates	NaN	NaN
United Kingdom	32.0	260.0
Unspecified	NaN	NaN

InvoiceDate	2011-12-09 12:09:00	2011-12-09 12:16:00 \
Country		
Australia	NaN	NaN
Austria	NaN	NaN
Bahrain	NaN	NaN
Belgium	NaN	NaN
Brazil	NaN	NaN
Canada	NaN	NaN
Channel Islands	NaN	NaN
Cyprus	NaN	NaN
Czech Republic	NaN	NaN
Denmark	NaN	NaN
EIRE	NaN	NaN
European Community	NaN	NaN
Finland	NaN	NaN
France	448.0	NaN
Germany	NaN	NaN
Greece	NaN	NaN
Hong Kong	NaN	NaN
Iceland	NaN	NaN
Ireland	NaN	NaN
Israel	NaN	NaN
Italy	NaN	NaN
Japan	NaN	NaN
Lebanon	NaN	NaN
Lithuania	NaN	NaN
Malta	NaN	NaN
Netherlands	NaN	NaN
Norway	NaN	NaN
Poland	NaN	NaN
Portugal	NaN	NaN
RSA	NaN	NaN
Saudi Arabia	NaN	NaN
Singapore	NaN	NaN
Spain	NaN	NaN
Sweden	NaN	NaN
Switzerland	NaN	NaN
USA	NaN	NaN
United Arab Emirates	NaN	NaN
United Kingdom	32.0	260.0
Unspecified	NaN	NaN

InvoiceDate	2011-12-09 12:21:00	2011-12-09 12:23:00 \
Country		
Australia	NaN</	