import numpy as np  
import pandas as pd  
Customer\_df=pd.read\_csv("https://raw.githubusercontent.com/swapnilsaurav/OnlineRetail/master/customers.csv")  
*#print("Customer deatils:\n",Customer\_df)*Order\_df=pd.read\_csv("https://raw.githubusercontent.com/swapnilsaurav/OnlineRetail/master/order\_items.csv")  
*#print("Order details:\n",Order\_df)  
#display all the column names*print(list(Order\_df.columns))  
*#required columns*Order\_df=Order\_df[['order\_id','product\_id','price']]  
print("Order details:\n",Order\_df)  
product\_df=pd.read\_csv("https://raw.githubusercontent.com/swapnilsaurav/OnlineRetail/master/products.csv")  
*#print("Product details:\n",product\_df)  
#display all the column names*print(list(product\_df.columns))  
*#required columns*product\_df=product\_df[['product\_id','product\_category\_name']]  
print("Product details:\n",product\_df)  
*#Read category translation file*cat\_df =pd.read\_csv('https://raw.githubusercontent.com/swapnilsaurav/OnlineRetail/master/product\_category\_name.csv')  
*#Display all the column names*print (list (cat\_df.columns))  
*#Let's rename the column names*cat\_df = cat\_df.rename(columns={  
 '1 product\_category\_name': 'product\_category\_name',  
 '2 product\_category\_name\_english': 'product\_category'})  
print (list (cat\_df.columns))  
*#Final dataset - merge tab 1 and tab2*data =pd.merge (Order\_df, product\_df, on='product\_id',how='left')  
*#Now merge with category to get English category*data =pd.merge (data, cat\_df,  
on='product\_category\_name', how='left')  
*#Check for Missing Data Percentage List*for col in data.columns:  
 pct\_missing = np.mean (data [col].isnull())  
 print('{} - {}%'.format(col, round (pct\_missing\*100)))  
  
 *# product\_category\_name-1 % lets create a new category called a new category called Unknown data* data['product\_category']=data['product\_category'].fillna("Unknown")  
 *# Check if all rows have been accounted for  
 # if not then merge didnt happen correctly* print("Number of rows: \n\n order\_items [{}],\n\n MergedData[{}]".format (Order\_df.count(), data.count()))  
 *# Note: Number of rows in order\_items and MergedData should be same  
 # if you want to push the content to a csv file and  
 # perform manual test, then uncomment the below line* data.to\_csv("TestingMergel.csv")  
 *# We are now ready to perform the Pareto analysis*import seaborn as sns  
import matplotlib.pyplot as plt  
from matplotlib.ticker import PercentFormatter  
df=data[['price', 'product\_category']]  
df.set\_index (data['product\_category'])  
*#Initially test with small dataset to see what you get df.head (100) #review with smaller dataset*print(df.head(100))  
*#Analysis 1: What is the most in demand product category?*sns.countplot (df ['product\_category'], order =  
df ['product\_category'].value\_counts().index)  
plt.title('Product Categories based on Demand'.title(), fontsize=20)  
plt.ylabel('count'.title(), fontsize=14)  
plt.xlabel('product category'.title(), fontsize=14)  
plt.xticks(rotation=90, fontsize=10)  
plt.yticks (fontsize=12)  
plt.close('all')  
plt.show()  
*#2: Which categories generates high sales-Pareto  
# Sort the values in the descending order*quant\_variable = df ['price']  
by\_variable =df['product\_category']  
column ='price'  
group\_by ='product\_category'  
df = df.groupby(group\_by) [column].sum().reset\_index()  
df = df.sort\_values (by=column, ascending=False)  
df ["cumpercentage"]=df[column].cumsum()/df[column].sum()\*100  
fig, ax=plt.subplots(figsize=(20,5))  
ax.bar(df['product\_category'], df['price'], color="C0")  
ax2 = ax.twinx ()  
ax2.plot(df[group\_by], df ["cumpercentage"], color="C1",  
marker="D", ms=7)  
ax2.yaxis.set\_major\_formatter (PercentFormatter ())  
ax.tick\_params(axis="x", rotation=90)  
ax.tick\_params(axis="y", colors="C0")  
ax2.tick\_params(axis="y", colors="C1")  
plt.title('Product Categories based on Sales'.title(),  
fontsize=20)  
*#plt.close('all')*plt.show()  
*#Variation 2  
#Plotting above graph with only top 40 categories, rest as Other categories*total=quant\_variable.sum()  
df =df.groupby(group\_by) [column].sum().reset\_index ()  
df = df.sort\_values (by=column, ascending=False)  
df ["cumpercentage"]=df [column]. cumsum()/df [column].sum()\*100  
threshold = df [column].cumsum() /5 *#20%*import pandas as pd  
import matplotlib.pyplot as plt  
from matplotlib.ticker import PercentFormatter  
*# Filter the DataFrame based on the threshold*df\_above\_threshold = df[df['cumpercentage'] < threshold]  
df\_below\_threshold = df[df['cumpercentage'] >= threshold]  
*# Calculate the sum total of the specified column*sum\_total = df[column].sum()  
*# Calculate the remaining cumulative percentage*restbarcumsum = 100 - df\_above\_threshold['cumpercentage'].max()  
*# Create a Series for the "OTHERS" category*rest = pd.Series(['OTHERS', sum\_total, restbarcumsum], index=[group\_by, column, 'cumpercentage'])  
*# Append the "OTHERS" category to the DataFrame*df\_above\_threshold = pd.concat([df\_above\_threshold, rest.to\_frame().T], ignore\_index=True)*# Set the index to the group\_by column and sort by cumulative percentage*df.index = df[group\_by]  
df = df.sort\_values(by='cumpercentage', ascending=True)  
*# Create the bar plot with a secondary y-axis for cumulative percentage*fig, ax = plt.subplots()  
*# Bar plot for the specified column*ax.bar(df.index, df[column], color="C0")  
*# Secondary y-axis for cumulative percentage*ax2 = ax.twinx()  
ax2.plot(df.index, df["cumpercentage"], color="C1", marker="D", ms=7)  
*# Format the y-axis as percentages*ax2.yaxis.set\_major\_formatter(PercentFormatter())  
*# Set tick parameters for both axes*ax.tick\_params(axis="x", colors="C0", labelrotation=90)  
ax.tick\_params(axis="y", colors="C0")  
ax2.tick\_params(axis="y", colors="C1")  
*# Set the plot title*plt.title('Product Categories Based on Sales'.title(),fontsize=20)  
*# Show the plot*plt.show()