Ecommerce Sentiment Analysis and Review Processing Using Python and SQL

INTRODUCTION:

The project aims to conduct a comprehensive analysis of an e-commerce platform using the provided dataset. The dataset contains information about the products, customer reviews, purchase history, seller details, categories, and other relevant details. The project seeks to uncover valuable insights about customer behavior, popular products, seller performance, customer satisfaction, and overall platform performance through various data analysis techniques.

DATA WRANGLING

In [1]:
```
#import the necessary libraries

import sqlite3
import numpy as np
import pandas as pd
```

In [3]:
```
#import a self define module

import settings #the codes in this modules will be posted at the end of this file
```

In [4]:
```
#import the toolkits for the sentiment analysis

import nltk
nltk.download()
from nltk.sentiment import SentimentIntensityAnalyzer
```

In [5]:
```
#load the uncleaned dataset
df = pd.read_csv('shopping.csv')
```

In [6]:
```
#print head of the dataset
df.head(2)
```

Out[6]:

	name	brand	categories	primaryCategories	reviews.date	reviews.text	reviews.title
0	All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi...	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	Electronics	2016-12-26T00:00:00.000Z	Purchased on Black FridayPros - Great Price (e...	Powerful tablet
1	Amazon - Echo Plus w/ Built-In Hub - Silver	Amazon	Amazon Echo,Smart Home,Networking,Home & Tools...	Electronics,Hardware	2018-01-17T00:00:00.000Z	I purchased two Amazon in Echo Plus and two do...	Amazon Echo Plus AWESOME

In [7]:
```
#check for any duplicates and sum it to know the amount of duplicates present in the dataset
df.duplicated().sum()
```

Out[7]:
```
28
```

In [8]:
```
#drop the duplicates
df.drop_duplicates(inplace=True)
```

In [9]:
```
#check for thr null values in each column
df.isnull().sum()
```

Out[9]:
```
name                  0
brand                 0
categories            0
primaryCategories     0
reviews.date          0
reviews.text          0
reviews.title        10
dtype: int64
```

In [10]:
```
#drop the null values
df.dropna(inplace=True)
```

In [11]:
```
#rename the messy columns
df = df.rename( columns={'reviews.text': 'reviews_text',
                         'reviews.title': 'reviews_title',
                         'reviews.date': 'reviews_date'})
```

In [12]:
```
#check the info about the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3962 entries, 0 to 3999
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   name                3962 non-null   object
 1   brand               3962 non-null   object
 2   categories          3962 non-null   object
 3   primaryCategories   3962 non-null   object
 4   reviews_date        3962 non-null   object
 5   reviews_text        3962 non-null   object
 6   reviews_title       3962 non-null   object
dtypes: object(7)
memory usage: 247.6+ KB
```

In [13]:
```
#print again the head of the dataset
df.head(2)
```

Out[13]:

	name	brand	categories	primaryCategories	reviews_date	reviews_text	reviews_title
0	All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi...	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	Electronics	2016-12-26T00:00:00.000Z	Purchased on Black FridayPros - Great Price (e...	Powerful tablet
1	Amazon - Echo Plus w/ Built-In Hub - Silver	Amazon	Amazon Echo,Smart Home,Networking,Home & Tools...	Electronics,Hardware	2018-01-17T00:00:00.000Z	I purchased two Amazon in Echo Plus and two do...	Amazon Echo Plus AWESOME

In [14]:
```
#make a copy of the cleaned dataset
df1 = df.copy()
```

In [15]:
```
#create an object called sentiments
sentiments = SentimentIntensityAnalyzer()
```

In [16]:
```
# Create a compound column and sentiment column
df1['sentiment'] = df1['reviews_text'].apply(lambda x: sentiments.polarity_scores(x))
# Calculate sentiment scores for each review and assign 'positive' or 'negative' based on the scores
df1['compound'] = df1['sentiment'].apply(lambda d: d['neg'] - d['pos'])

df1['sentiment'] = df1['compound'].apply(lambda score: 'positive' if score < 0 else 'negative' )
```

In [17]:
```
#print head of the dataset
df1.head(2)
```

Out[17]:

	name	brand	categories	primaryCategories	reviews_date	reviews_text	reviews_title	sentiment	compound
0	All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi...	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	Electronics	2016-12-26T00:00:00.000Z	Purchased on Black FridayPros - Great Price (e...	Powerful tablet	positive	-0.280
1	Amazon - Echo Plus w/ Built-In Hub - Silver	Amazon	Amazon Echo,Smart Home,Networking,Home & Tools...	Electronics,Hardware	2018-01-17T00:00:00.000Z	I purchased two Amazon in Echo Plus and two do...	Amazon Echo Plus AWESOME	positive	-0.181

In [ ]:

```
# Tokenize the reviews_text column using nltk's word_tokenize function
df1['reviews_text'] = df1['reviews_text'].apply(nltk.word_tokenize)
# Normalize the tokenized words using the normalize function from settings
df1['reviews_text'] = df1['reviews_text'].apply(settings.normalize)
```

In [20]:

```
df1.head(2)
```

Out[20]:

	name	brand	categories	primaryCategories	reviews_date	reviews_text	reviews_title	sentiment	compound
0	All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi...	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	Electronics	2016-12-26T00:00:00.000Z	[Purchased, on, Black, FridayPros, -, Great, P...	Powerful tablet	positive	-0.280
1	Amazon - Echo Plus w/ Built-In Hub - Silver	Amazon	Amazon Echo,Smart Home,Networking,Home & Tools...	Electronics,Hardware	2018-01-17T00:00:00.000Z	[I, purchased, two, Amazon, in, Echo, Plus, an...	Amazon Echo Plus AWESOME	positive	-0.181

In [63]:

```
# Export the cleaned dataset to a new CSV file named 'ecommerce.csv'. use index = False.
df1.to_csv('ecommerce.csv', index = False)
```

In [53]:

```
cnn = sqlite3.connect('jupyter_sql_tutorial.db')
```

In [ ]:

```
#load the prepared dataset into sqlite database
df1.to_sql('ecommerce', cnn)
```

```
In [55]: %load_ext sql
```

```
The sql extension is already loaded. To reload it, use:
  %reload_ext sql
```

```
In [42]: %sql sqlite:///jupyter_sql_tutorial.db
```

```
In [105…  %%sql

          #check if all columns are correctly loaded
          SELECT *
          FROM ecommerce
          LIMIT 1;
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[105…

index	name	brand	categories	primaryCategories	reviews_date	reviews_text	reviews_title	sentiment
0	All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi, 16 GB - Includes Special Offers, Magenta	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Tablets,Tablets,Computers & Tablets	Electronics	2016-12-26T00:00:00.000Z	['purchased', 'on', 'black', 'fridaypros', 'great', 'price', 'even', 'off', 'sale', 'very', 'powerful', 'and', 'fast', 'with', 'quad', 'core', 'processors', 'amazing', 'soundwell', 'builtcons', 'amazon', 'ads', 'amazon', 'need', 'this', 'to', 'subsidize', 'the', 'tablet', 'and', 'will', 'remove', 'the', 'adds', 'if', 'you', 'pay', 'them', 'inability', 'to', 'access', 'other', 'apps', 'except', 'the', 'ones', 'from', 'amazon', 'there', 'is', 'a', 'way', 'which', 'i', 'was', 'able', 'to', 'accomplish', 'to', 'add', 'the', 'google', 'play', 'storenet', 'this', 'is', 'a', 'great', 'tablet', 'for', 'the', 'money']	Powerful tablet	positive

TASK 1: How many values are there in the given dataset

```
In [69]: %%sql
```

```
SELECT COUNT(*) as total_values
FROM ecommerce
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[69]:

total_values
3962

TASK 2: Find out the unique brands in the given dataset

In [72]:

```
%%sql

SELECT DISTINCT(brand) as unique_brands
FROM ecommerce
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[72]:

unique_brands
Amazon
Flipkart

TASK 3: Retrieve all records from the 'ecommerce' table where the brand is 'Amazon'.

In [73]:

```
%%sql

SELECT count(*) as total_records_from_amazon
FROM ecommerce
WHERE brand = 'Amazon'
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[73]:

total_records_from_amazon
2476

TASK 4: Retrieve all records from the 'ecommerce' table where the product reviews contain the word 'good' in their text.

In [77]:
```
%%sql

SELECT count(*) as total_records_with_good_in_their_review_text
FROM ecommerce
WHERE ecommerce.reviews_text LIKE '%good%'
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[77]:

total_records_with_good_in_their_review_text
481

TASK 5: Provide a list of all products and their corresponding details from the 'ecommerce' table that belong to the 'Electronics' category.

In [79]:
```
%%sql

SELECT count(*) as total_products_from_electronics
FROM ecommerce
WHERE categories LIKE '%Electronics%'
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[79]:

total_products_from_electronics
2436

TASK 6: Retrieve all records from the 'ecommerce' table where the products are categorized under 'Electronics' only as their primary category and the brand is 'Flipkart'.

In [81]:
```
%%sql

SELECT count(*) as total
FROM ecommerce
WHERE primaryCategories LIKE 'Electronics' AND brand LIKE 'Flipkart'
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[81]:

total
943

TASK 7: Provide a summary of the number of positive and negative sentiments for each primary category in the 'ecommerce' table.

In [85]:

```
%%sql

SELECT x.primaryCategories, IFNULL(x.positive_count,0) as positive_count, IFNULL(y.negative_count, 0) as negative_count
FROM (SELECT primaryCategories, COUNT(sentiment) as sentiment_count
      FROM ecommerce
      WHERE sentiment = 'sentiment'
      GROUP BY 1) as z

LEFT OUTER JOIN (SELECT primaryCategories, COUNT(sentiment) as positive_count
      FROM ecommerce
      WHERE sentiment = 'positive'
      GROUP BY 1) as x

LEFT OUTER JOIN (SELECT primaryCategories, COUNT(sentiment) as negative_count
      FROM ecommerce
      WHERE sentiment = 'negative'
      GROUP BY 1) as y

ON y.primaryCategories = z.primaryCategories

UNION

SELECT x.primaryCategories, IFNULL(x.positive_count,0) as positive_count, IFNULL(y.negative_count, 0) as negative_count
FROM (SELECT primaryCategories, COUNT(sentiment) as positive_count
      FROM ecommerce
      WHERE sentiment = 'positive'
      GROUP BY 1) as x

LEFT OUTER JOIN (SELECT primaryCategories, COUNT(sentiment) as negative_count
      FROM ecommerce
      WHERE sentiment = 'negative'
      GROUP BY 1) as y

ON x.primaryCategories = y.primaryCategories

LEFT OUTER JOIN (SELECT primaryCategories, COUNT(sentiment) as sentiment_count
      FROM ecommerce
      WHERE sentiment = 'sentiment'
      GROUP BY 1) as z

ON y.primaryCategories = z.primaryCategories
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[85]:

primaryCategories	positive_count	negative_count
Electronics	2304	278
Electronics,Hardware	1064	86
Electronics,Media	17	0
Office Supplies,Electronics	188	25

TASK 8: Retrieve all records from the 'ecommerce' table where the sentiment in the product reviews is classified as 'positive'.

In [87]:

```
%%sql

SELECT count(*) as total_products_with_positive_review
FROM ecommerce
WHERE sentiment = 'positive'
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[87]:

total_products_with_positive_review
3573

TASK 9: Provide a summary report for each brand in the 'ecommerce' table, including the total number of positive and negative sentiments in product reviews, the total number of reviews, and the percentage of positive and negative sentiments for each brand.

In [100…

```
%%sql

WITH T1 AS (
    SELECT
        brand,
        SUM(sentiment='positive') as total_pos_count,
        SUM(sentiment='negative') as total_neg_count,
        SUM(sentiment='sentiment') as total_sent_count,
        COUNT(sentiment) as review_no
    FROM ecommerce
    GROUP BY 1)

SELECT  IFNULL(T1.brand,'brand') as brand,
        IFNULL(T1.total_pos_count,0) as total_pos_count,
        IFNULL(T1.total_neg_count,0) as total_neg_count,
        IFNULL(T1.review_no,0) as review_no,
        ROUND(((total_pos_count * 1.0) / review_no *100),2) as pos_percentage,
        ROUND(((total_neg_count * 1.0) / review_no *100),2) as neg_percentage

FROM T1
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[100…

brand	total_pos_count	total_neg_count	review_no	pos_percentage	neg_percentage
Amazon	2227	249	2476	89.94	10.06
Flipkart	1346	140	1486	90.58	9.42

TASK 10: Retrieve a count of products for each primary category in the 'ecommerce' table

In [102…

```
%%sql

SELECT primaryCategories, COUNT(*) as total_product
FROM ecommerce
GROUP BY 1
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[102…

primaryCategories	total_product
Electronics	2582
Electronics,Hardware	1150
Electronics,Media	17
Office Supplies,Electronics	213

TASK 11: Retrieve all records from the 'ecommerce' table where the product name contains the word 'Tablet' as a substring

In [103…
```
%%sql

SELECT count(*) as total_products
FROm ecommerce
WHERE ecommerce.name LIKE 'Tablet%' OR ecommerce.name LIKE '%Tablet%' OR ecommerce.name LIKE '%Tablet'
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[103…

total_products
2301

TASK 12: Count the number of product reviews in the 'ecommerce' table where the text contains the word 'Alexa' as a substring.

In [104…
```
%%sql

SELECT COUNT(*) as total_product
FROm ecommerce
WHERE reviews_text LIKE 'Alexa%' OR reviews_text LIKE '%Alexa%' OR reviews_text LIKE '%Alexa'
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[104…

total_product
339