Deloitte Virtual Internship – Task 2

By Uditi Mehta

Code I Used:

```
# Step 1: Load Required Libraries and Dataset
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv('QVI_data.csv')

# Convert 'date' column to datetime format
df['date'] = pd.to_datetime(df['date'])

# Extract month from date
df['month'] = df['date'].dt.to_period('M')
# Step 2: Aggregate Monthly Metrics Per Store
monthly_data = df.groupby(['month', 'store_id']).agg({
    'sales_value': 'sum',
    'customer_id': 'nunique',
    'transaction_id': 'nunique'
}).reset_index()

# Calculate average number of transactions per customer
monthly_data['avg_transactions_per_customer'] = (
    monthly_data['transaction_id'] / monthly_data['customer_id']
)

# Step 3: Define Function to Find Similar Control Stores
def calculate_similarity(trial_store_id, metric='sales_value', months=None):
    trial_data = monthly_data[monthly_data['store_id'] == trial_store_id]
    if months:
        trial_data = trial_data[trial_data['month'].isin(months)]

    similarities = []
    for store_id in monthly_data['store_id'].unique():
        if store_id == trial_store_id:
            continue

        control_data = monthly_data[monthly_data['store_id'] == store_id]
        if months:
```

```
            control_data = control_data[control_data['month'].isin(months)]

        # Merge on month to align trial and control store data
        merged = pd.merge(trial_data, control_data, on='month', suffixes=('_trial', '_control'))
        if merged.empty:
            continue

        # Calculate Pearson correlation
        corr = merged[f'{metric}_trial'].corr(merged[f'{metric}_control'])
        similarities.append((store_id, corr))

    # Sort and return highest correlation values
    return sorted(similarities, key=lambda x: -x[1])

# Step 4: Run Similarity Matching for Trial Stores
trial_stores = [77, 86, 88]

# Get top 3 similar control stores for each trial store
for store in trial_stores:
    print(f"\nTop control store matches for Trial Store {store}:")
    matches = calculate_similarity(
        trial_store_id=store,
        metric='sales_value',
        months=monthly_data['month'].unique()
    )
    for store_id, similarity in matches[:3]:
        print(f"Control Store {store_id} → Similarity Score: {similarity:.2f}")
```

Summary:

In this analysis, I evaluated the performance of stores 77, 86, and 88 by comparing them to control stores with similar historical sales patterns.

Using monthly aggregated sales and customer data, I implemented a Pearson correlation-based similarity metric to identify the best-matching control stores for each trial store.

This approach lays the foundation for uplift analysis and helps assess the effectiveness of trial interventions in improving store performance.

Tools Used: Python (Pandas, NumPy, Matplotlib)