

Experiment 5: Develop Competitive Intelligence Analysis programs for analysis of keyword analysis.

Aim

To analyze competitive keyword performance using Google Trends, and estimate user clicks, conversions, and return on investment (ROI) using a simulated digital marketing funnel.

Objectives

- Fetch real-time keyword interest data using the PyTrends API
- Calculate Share of Shelf (average search interest) for each keyword
- Simulate user behavior using Click-Through Rate (CTR) mapping
- Estimate conversions and sales based on fixed conversion rates and ASP
- Compute ROI based on assumed advertising costs
- Present a final summary table with actionable business insights

Pseudocode

```
# Import necessary libraries
from pytrends.request import TrendReq
import pandas as pd

# Initialize pytrends connection
pytrends = TrendReq hl='en-IN', tz=330)

# Define keywords for competitive analysis
kw1 = "Swiggy"
kw2 = "Zomato"
keywords = [kw1, kw2]

# Build payload and fetch interest over time data
pytrends.build_payload(keywords, geo='IN', timeframe='today 12-m')
df = pytrends.interest_over_time()

# Drop incomplete rows
if 'isPartial' in df.columns:
    df = df.drop(columns='isPartial')

# Calculate average interest score (Share of Shelf)
share = df.mean().round(2)
```

```
# Rank keywords based on average interest
ranked = share.sort_values(ascending=False).reset_index()
ranked.columns = ['Keyword', 'Share_on_Shelf']
ranked['rank'] = ranked.index + 1

# Map rank to click-through rate (CTR)
ctr_map = {
    1: 0.30, 2: 0.17, 3: 0.12, 4: 0.10, 5: 0.08,
    6: 0.07, 7: 0.06, 8: 0.05, 9: 0.04, 10: 0.03
}
ranked["ctr"] = ranked["rank"].map(ctr_map).fillna(0.02)

# Simulate impressions
ranked["impressions"] = 10000

# Calculate clicks and conversions
ranked["clicks"] = ranked["impressions"] * ranked["ctr"]
ranked["conversion_rate"] = 0.15
ranked["conversions"] = ranked["clicks"] * ranked["conversion_rate"]

# Set average selling price (ASP) and calculate sales
ranked["asp"] = 300
ranked["sales"] = ranked["conversions"] * ranked["asp"]

# Calculate ad spend and return on investment (ROI)
ranked["cpc"] = 6
ranked["ad_spend"] = ranked["clicks"] * ranked["cpc"]
ranked["roi"] = ranked["sales"] / ranked["ad_spend"]
```

Full Program

```
# Install pytrends (used to fetch Google Trends data)
!pip install pytrends --quiet

# Import required libraries
from pytrends.request import TrendReq
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Initialize pytrends session
pytrends = TrendReq hl='en-IN', tz=330)

# Define two competing keywords for analysis
```

```
kw1 = "Swiggy"
kw2 = "Zomato"
keywords = [kw1, kw2]

# Fetch search interest data for the last 12 months in India
pytrends.build_payload(keywords, geo='IN', timeframe='today 12-m')
df = pytrends.interest_over_time()

# Drop 'isPartial' column if present to clean up the dataset
if 'isPartial' in df.columns:
    df = df.drop(columns='isPartial')

# Plot the interest trends over time to visualize which keyword had more interest
plt.figure(figsize=(10, 5))
sns.lineplot(data=df)
plt.title(f"Interest Over Time: {kw1} vs {kw2}")
plt.xlabel("Date")
plt.ylabel("Search Interest")
plt.grid(True)
plt.tight_layout()
plt.show()

# Calculate average interest over the time period
# This represents the 'Share of Shelf' - how much attention each brand gets in search
share = df.mean().round(2)
print("Share of Shelf (Avg Search Interest):")
print(share)

# Rank the keywords based on average interest (higher interest gets higher rank)
ranked = share.sort_values(ascending=False).reset_index()
ranked.columns = ['Keyword', 'Share_on_Shelf']
ranked['rank'] = ranked.index + 1

# Map rank to expected Click Through Rate (CTR) using industry-standard estimate
# Higher ranked search results typically receive a higher percentage of clicks
ctr_map = {
    1: 0.30, 2: 0.17, 3: 0.12, 4: 0.10, 5: 0.08,
    6: 0.07, 7: 0.06, 8: 0.05, 9: 0.04, 10: 0.03
}
ranked["ctr"] = ranked["rank"].map(ctr_map).fillna(0.02) # Use fallback CTR for

# Simulate a search volume of 10,000 impressions for each keyword
ranked["impressions"] = 10000

# Estimate total clicks each keyword receives using CTR
ranked["clicks"] = ranked["impressions"] * ranked["ctr"]
```

```
# Assume a 15% conversion rate (i.e., 15% of users who click will complete a purchase)
ranked["conversion_rate"] = 0.15
ranked["conversions"] = ranked["clicks"] * ranked["conversion_rate"]

# Set an average selling price (ASP) of ₹300 per conversion
ranked["asp"] = 300
ranked["sales"] = ranked["conversions"] * ranked["asp"]

# Assume cost-per-click (CPC) is ₹6 for each paid click
ranked["cpc"] = 6
ranked["ad_spend"] = ranked["clicks"] * ranked["cpc"]

# Calculate ROI: Return on Investment = Revenue / Ad Spend
ranked["roi"] = ranked["sales"] / ranked["ad_spend"]

# Display a simplified summary table with key insights
summary = ranked[["Keyword", "Share_on_Shelf", "conversions", "roi"]].round(2)

# Show the final result
print("\nSimulated Keyword Conversion and ROI Summary:")
display(summary)
```