Amazon Sales VS Google Search Interests

Bus 312

Jeff Song, Sujas Nahar, Navya Kakkar, Camila Medina

Research Question & Hypothesis

Primary Question

How does Google search interest correlate with Amazon sales revenue?

Hypothesis

Higher search interest indicates stronger consumer demand and predicts sales performance.

Scope

Analysis across multiple product categories over time.

Data Sources Overview

This analysis combines two primary data sources:

Dataset 1 — Amazon.csv

- Transactions: orders, products, categories
- Columns: UnitPrice, Quantity, Discount, Tax, ShippingCost

Dataset 2- **Google Trends Data (API)**

Using PyTrends, we retrieved weekly search interest values for the **top 5 Amazon categories**. The timeframe was synchronized with the Amazon dataset.

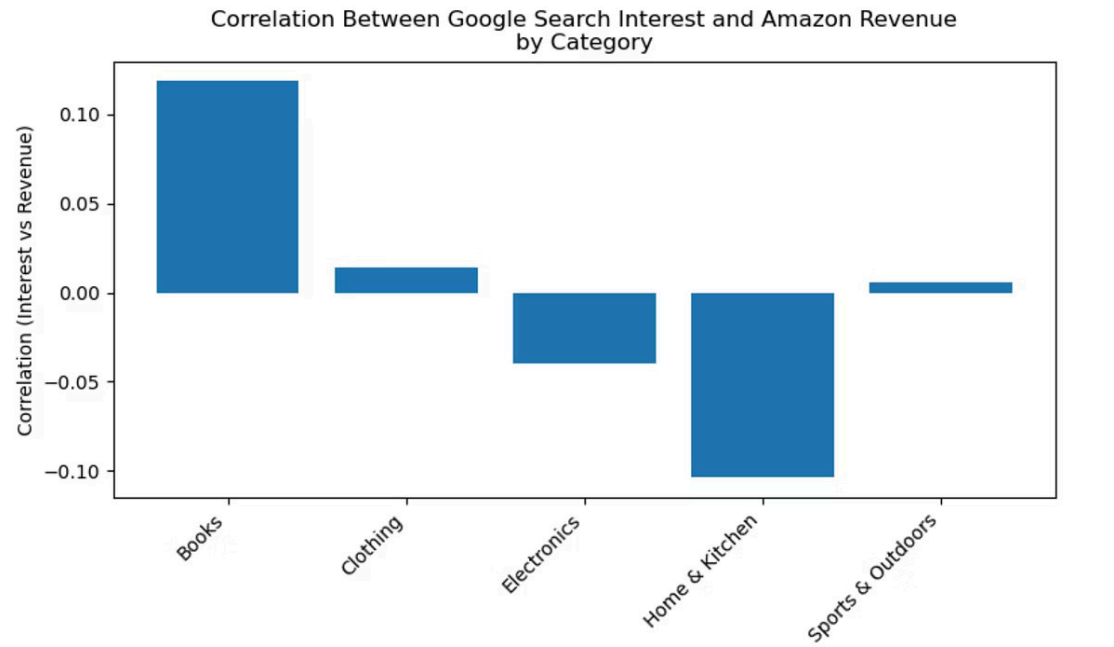
Google Trends provides normalized search volume (0–100), representing consumer interest over time.

Data Preparation & Cleaning

Before analysis, we cleaned and prepared both datasets: removed duplicate entries, handled missing values in sales records, standardized date formats across both APIs, normalized search interest scores (0-100 scale), and aligned data by product category and time period. Tools used: Python (pandas, numpy) for data manipulation and cleaning.

- Removed whitespace/empty strings
- Converted types (dates, numerics)
- Dropped invalid rows (missing Quantity, OrderID, etc.)
- Removed duplicate orders
- Fixed inconsistent TotalAmount by recalculating
- Filtered to delivered orders
- Added OrderYear, OrderMonth, YearMonth

```python
def clean_amazon_orders(df: pd.DataFrame) -> pd.DataFrame:
    cleaned = df.copy()

    # Strip whitespace from all object columns
    obj_cols = cleaned.select_dtypes(include="object").columns
    cleaned[obj_cols] = cleaned[obj_cols].apply(lambda col: col.str.strip())

    # Empty strings -> NaN
    cleaned.replace({"": np.nan, " ": np.nan}, inplace=True)

    # Parse dates
    cleaned["OrderDate"] = pd.to_datetime(cleaned["OrderDate"], errors="coerce")

    # Numeric conversions
    numeric_cols = ["Quantity", "UnitPrice", "Discount", "Tax", "ShippingCost", "TotalAmount"]
    for col in numeric_cols:
        cleaned[col] = pd.to_numeric(cleaned[col], errors="coerce")

    # Drop rows missing critical fields
    critical_cols = ["OrderID", "OrderDate", "CustomerID", "ProductID", "Quantity", "UnitPrice"]
    cleaned = cleaned.dropna(subset=critical_cols)

    # Fill some non-critical text columns with "Unknown"
    fill_defaults = {
        "CustomerName": "Unknown",
        "Category": "Unknown",
        "Brand": "Unknown",
        "PaymentMethod": "Unknown",
        "OrderStatus": "Unknown",
        "City": "Unknown",
        "State": "Unknown",
        "Country": "Unknown",
        "SellerID": "Unknown",
    }
    for col, default in fill_defaults.items():
        if col in cleaned.columns:
            cleaned[col] = cleaned[col].fillna(default)

    # Drop exact duplicate rows
    before_dups = cleaned.shape[0]
    cleaned = cleaned.drop_duplicates()
    after_dups = cleaned.shape[0]
    print(f"Rows removed as exact duplicates: {before_dups - after_dups}")

    # Drop duplicate OrderIDs (keep first)
    before_order_ids = cleaned.shape[0]
    cleaned = cleaned.drop_duplicates(subset="OrderID", keep="first")
    after_order_ids = cleaned.shape[0]
    print(f"Rows removed due to duplicate OrderID: {before_order_ids - after_order_ids}")

    # Basic sanity filters
    cleaned = cleaned[cleaned["Quantity"] > 0]
    cleaned = cleaned[cleaned["UnitPrice"] >= 0]
    cleaned = cleaned[cleaned["Tax"] >= 0]
    cleaned = cleaned[cleaned["ShippingCost"] >= 0]
    cleaned = cleaned[cleaned["Discount"].between(0, 1)]

    # Recalculate TotalAmount where it's inconsistent
    calc_total = (
        cleaned["Quantity"] * cleaned["UnitPrice"] * (1 - cleaned["Discount"])
        + cleaned["Tax"]
        + cleaned["ShippingCost"]
    )
    diff = (cleaned["TotalAmount"] - calc_total).abs()
    mismatches = (diff > 0.01).sum()
    print(f"Rows with inconsistent TotalAmount: {mismatches}")

    # Overwrite TotalAmount if mismatch is big
    cleaned.loc[diff > 0.01, "TotalAmount"] = calc_total.round(2)

    # Time features
    cleaned["OrderYear"] = cleaned["OrderDate"].dt.year
    cleaned["OrderMonth"] = cleaned["OrderDate"].dt.to_period("M")   # monthly period
```

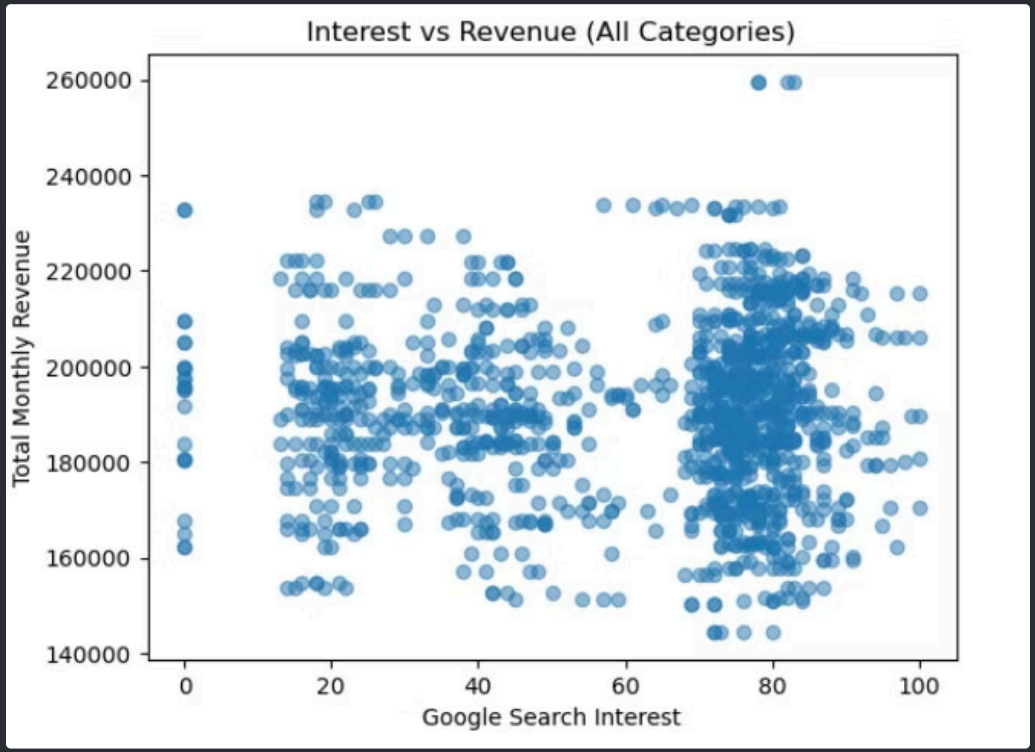


Correlation Between Google Search Interest and Amazon Revenue by Category

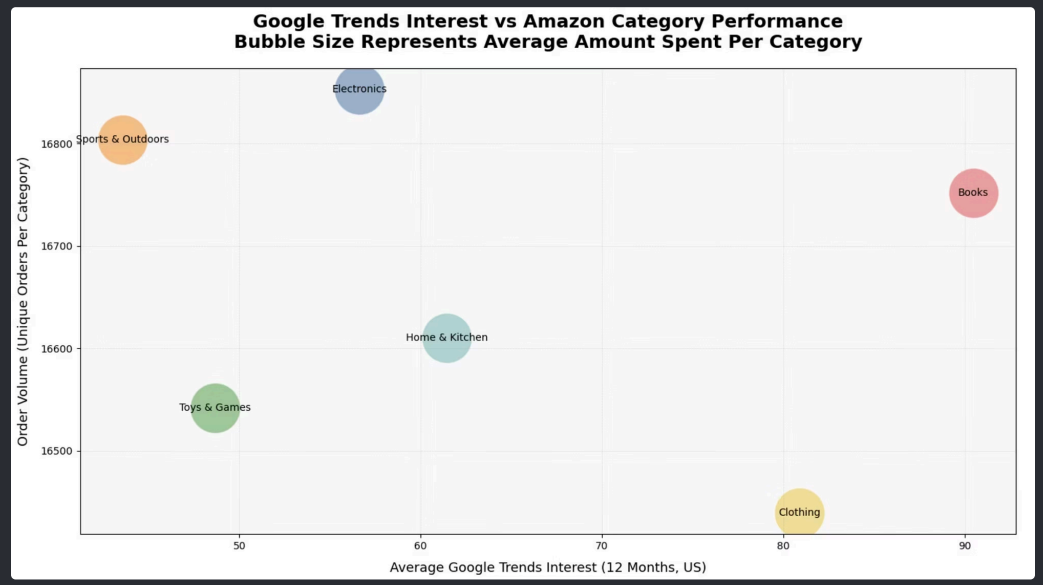
This chart illustrates the dynamic relationship between Google Search interest and Amazon monthly revenue by category from 2020 to 2025.

- Books show the strongest positive relationship between search interest and revenue
- Clothing and Sports & Outdoors have very weak or neutral correlations
- Electronics and Home & Kitchen show slightly negative correlations
- Overall, search interest does not consistently predict revenue across categories

Interest vs Revenue Across All Categories



- There is **no strong linear correlation** when combining all categories
- The scatterplot shows **distinct clusters**, meaning categories behave very differently
- There is a **slight upward trend**, but it is weak and inconsistent
- Higher search interest sometimes aligns with higher revenue, but **not reliably across all categories**
- **Conclusion:** Google Trends is helpful context, but **cannot predict Amazon revenue on its own** when categories are mixed



Category Specific Insights

1. High-Correlation Categories (Electronics, Home & Kitchen, etc.)

- These categories show strong search-to-sales alignment

2. Anomalies: Clothing & Sports/Outdoors

- Negative or weak correlation suggests search interest *doesn't* predict sales here

3. Volume vs. Spending Disconnect

- Similar average spending across categories but *very different* purchase volumes

Key Insights

1. **Search interest strongly aligns with sales** in most consumer-focused categories.
2. **Seasonal patterns** are visible and consistent.
3. **Lag analysis** suggests that search interest may slightly **precede** sales, offering predictive value.
4. Some categories are inherently noisy due to broader keyword definitions.