Basic Idea: a data-driven e-commerce intelligence project.

**Project Overview: Trend-Based Product and Pricing Recommendation**

**Goal**

Build a machine learning model that:

1. **Analyzes Instagram trending topics (e.g., via Reels).**
2. **Maps these topics to actual products.**
3. **Finds sellers and their prices (market research).**
4. **Predicts which product to sell and what price to sell it at** (slightly lower than competitors, still profitable).

**Step-by-Step Structure**

**1. Get Instagram Reels Trending Topics → Excel Sheet 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic** | **Popularity Score** | **Timestamp** | **Related Product** |
| Stanley Cup | 98 | 2025-07-01 | Stanley Tumbler |
| Pickle Skincare | 87 | 2025-07-01 | Pickle Serum |

**How to Automate This:**

Instagram doesn’t offer easy API access for reels trends, so:

* **Option 1: Scraping (Semi-legal workaround, be cautious)**
  + Use Playwright or Selenium to **scrape Instagram’s Explore page or Reels**.
  + Parse video captions, hashtags, and views.
  + Use NLP (e.g., spaCy, transformers) to extract popular topics.
* **Option 2: Use third-party APIs/tools**
  + **BuzzSumo**, **Trendinalia**, **Brand24**, **Ingramer**: These offer APIs or dashboards to track social trends.
  + Use Python (requests, pandas) to pull data and update an Excel sheet automatically.
* **Option 3: Manual dummy data for prototyping**
  + Create mock data for now, like 10–15 rows of “Topic”, “Hashtag popularity”, and “Related Product”.

**2. Market Research → Excel Sheet 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product** | **Seller** | **Platform** | **Price (INR)** | **Link** |
| Stanley Tumbler | Milton | Amazon | 999 | amazon.com/product-id |
| Stanley Tumbler | Cello | Flipkart | 849 | flipkart.com/product-id |

**How to Automate This:**

Use web scraping and APIs to collect real-time pricing:

* **Scraping Tools**:
  + Use BeautifulSoup or Scrapy in Python to extract product listings from:
    - Amazon
    - Flipkart
    - Myntra
    - Nykaa
  + Look for:
    - Title
    - Price
    - Seller Name
    - Product URL
* **APIs**:
  + Use **SerpAPI** (Google Shopping), **Octoparse**, or **Zenserp** to get product listings from search queries like "Stanley Cup site:flipkart.com".
* **Storage**:
  + Use pandas to write to Excel:
* **Scheduler**:
  + Use cron or APScheduler to run these scrapers daily or weekly and auto-update the Excel files.

**3. Machine Learning Model → Decision Engine**

|  |  |
| --- | --- |
| **Feature** | **Example** |
| Topic Popularity Score | 98 |
| Avg Market Price | 999 |
| Lowest Competitor Price | 849 |
| Suggested Selling Price | 799 |

**ML Model Goal**

Predict: "Which product to sell?" and "At what price?"

**Steps:**

1. **Feature Engineering**: Merge both sheets by matching topics to products.
2. **Target Variable**: Choose profitability or conversion likelihood (use dummy labels initially).
3. **Model**: Use regression/classification depending on approach:
   * Classification: Which product should I sell?
   * Regression: What price to sell it at?

**Starter Models**:

* Linear Regression
* Decision Trees
* Random Forests
* XGBoost

**🧪 MVP Roadmap**

**🔹 Phase 1 – Setup (Dummy Data)**

* Prepare **2 dummy Excel files**:
  1. Trending Topics + Popularity + Product
  2. Product + Seller + Price
* Merge them using pandas.
* Build a small ML model to predict a price 5–10% lower than the minimum price while maintaining a margin.

**🔹 Phase 2 – Automation**

* Scrape Instagram + Marketplaces.
* Auto-update Excel or directly store in database.
* Add scheduling (daily/weekly updates).

**🔹 Phase 3 – Advanced Modeling**

* Include more data like:
  + Number of Reels
  + Product reviews
  + Stock status
  + User location trends
* Improve model accuracy and profit estimations.

**Data Automation Suggestions**

|  |  |
| --- | --- |
| **Task** | **Tool/Library** |
| Instagram Reels Scraping | Playwright, Selenium, proxy use |
| Web Marketplace Scraping | BeautifulSoup, Scrapy |
| Google Shopping Results | SerpAPI, Zenserp |
| Scheduled Automation | cron, APScheduler |
| Excel File Handling | pandas, openpyxl |

[**https://github.com/Kiwi-Supreme/Adobe\_Hackathon**](https://github.com/Kiwi-Supreme/Adobe_Hackathon)