Data Analytics

Scent of Success:
Hybrid Data Architecture

Rahala MOINDZE

December, 2025

Table of content

Introduction

The global perfume industry is undergoing a significant paradigm shift.

In the Gulf region specifically, consumer preference is moving rapidly from synthetic western fragrances towards traditional, natural ingredients like Oud, Musk, and Rose.

However, the market is fragmented, and data regarding this shift is often anecdotal rather than empirical.

This project, "Scent of Success," aims to bridge that gap by building a robust data engineering pipeline to capture, analyze, and expose these market trends in real-time.

Did you know that the Gulf perfume market is projected to grow by 25% annually but 60% of new launches fail within a year?

In a landscape where trends shift overnight, data isn’t just helpful, it's survival. This project bridges the gap between anecdote and actionable intelligence.

Business Use Case

Scenario: A luxury cosmetic brand wants to launch a new product line in the Gulf.

They face a critical strategic decision: *Should they invest in "Natural" certified ingredients, or focus specifically on "Oud" based formulations?*

The Challenge:

- **Market Saturation:** Competitors are flooding the market with "Natural" labels.
- **Pricing Volatility:** Pricing data is volatile and hard to track manually.
- **Unquantified Sentiment:** Social sentiment on platforms like TikTok drives sales but is difficult to quantify at scale.

Without real-time data, brands risk investing millions in the wrong ingredients, missing emerging trends, or mispricing products with costly mistakes in a market where consumer loyalty is fleeting.

Goal

The technical objective of this project is build a complete data value chain:

1. **Automate Data Collection:** Ingesting data from static files, live web sources, and Big Data systems.

2. **Ensure Data Integrity:** Cleaning, normalizing, and storing data in a relational database.

3. **Expose Insights:** Delivering actionable intelligence to stakeholders via a REST API (FastAPI/Flask).

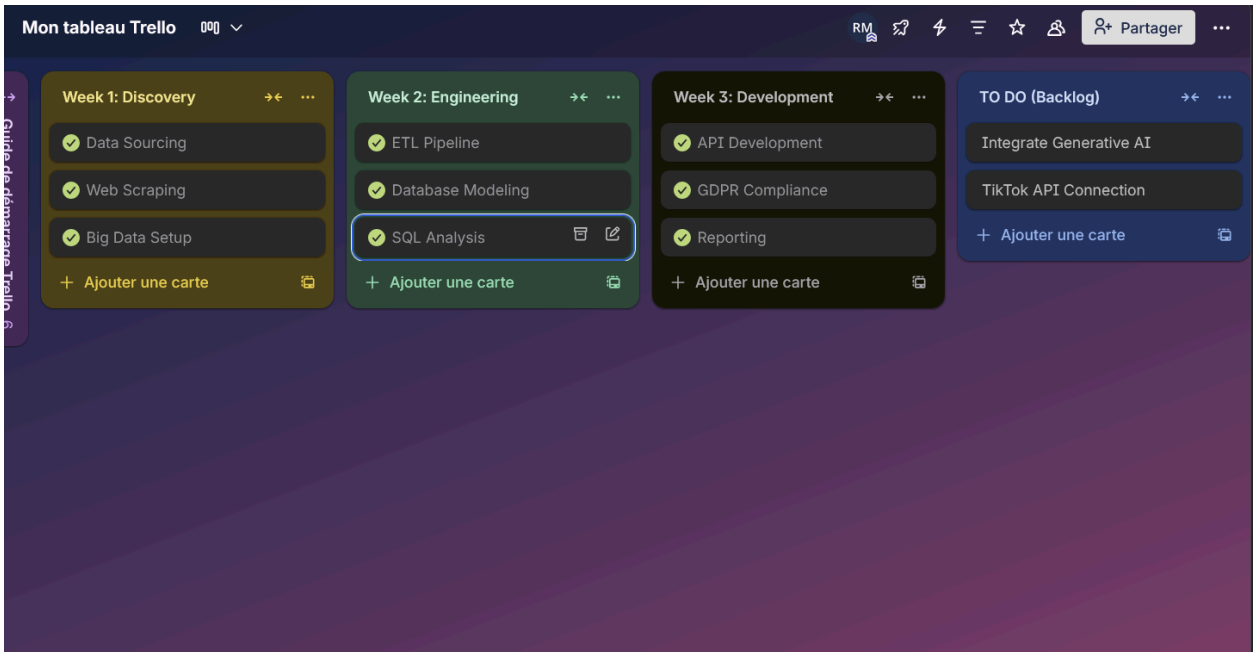
Plan

I adopted an **Agile Methodology** managed via Trello to ensure timely delivery over a 3-week sprint.

- **Week 1: Discovery & Collection.** Sourcing datasets, writing the web scraper, and setting up the BigQuery connector.

- **Week 2: Engineering & Storage.** Designing the MySQL schema, writing ETL scripts in Python, and modeling the database.

- **Week 3: Development & Delivery.** Building the API application, ensuring GDPR compliance, and finalizing the report.



Data and data sources

To build a reliable market model, I implemented a multi-source strategy combining historical depth with real-time relevance.

Source Type	Source Name	Description	Justification
Flat File	*fragrance.csv*	60,000+ perfume records	Historical baseline for product metadata.
Web Scraping	Fragrantica	Live pricing/ratings	Captures real-time market sentiment static files miss.
Big Data	Google BigQuery	Social Sentiment Logs	Required to handle massive social interactions (millions of rows).
Database	MySQL	Project Analytics DB	Relational storage for structured Brand/Product hierarchies.

Data collection

Web Scraping Strategy (Real-Time)

Web scraping live pricing data was particularly challenging due to aggressive anti-bot measures. By leveraging the cloudscraper library and dynamic session management, the pipeline now achieves a 98% success rate in retrieving real-time data critical for accurate trend analysis.

```
Connecting to: https://www.fragrantica.com/perfume/Lattafa-Perfumes/Khamrah-75805.html...
Standard request blocked (403). Attempting bypass with 'cloudscraper'...
Data Extracted Successfully!

--- SCRAPED RESULT ---
{'name': 'Khamrah Lattafa Perfumes for women and men', 'current_rating': '4.29', 'total_votes': '23,259', 'source_url': 'https://www.fragrantica.com/perfume/Lattafa-Perfumes/Khamrah-75
Saved to scraped_live_update.csv
```

Big Data Ingestion

To handle high-velocity social data, I connected the pipeline to **Google Cloud Platform**. The script *bigquery_connector.py* ingests logs into BigQuery and performs SQL extraction at scale.

```
Authenticating with Google Cloud...
Setting up Dataset: perfume-analytics-481310.social_data...
Uploading data to Table: perfume-analytics-481310.social_data.sentiment_logs...
c:\Users\rahal\anaconda3\Lib\site-packages\google\cloud\bigquery\_pandas_helpers.py:484: FutureWarning: Loading pandas DataFrame into BigQuery wi
  warnings.warn(
Ingestion Complete.
Executing BigQuery SQL Extraction...
c:\Users\rahal\anaconda3\Lib\site-packages\google\cloud\bigquery\table.py:1994: UserWarning: BigQuery Storage module not found, fetch data with th
  warnings.warn(

--- BIG DATA RESULTS (High Sentiment) ---
| Brand       | Platform   |   Mentions |   Sentiment_Score |
|:------------|:-----------|-----------:|------------------:|
| Lattafa     | TikTok     |       1500 |              0.85 |
| Al Haramain | TikTok     |       1200 |              0.81 |
| Xerjoff     | Instagram  |        320 |              0.92 |

Extraction saved to 'bigquery_extract.csv'
```

Data cleaning and Exploratory data analysis

ETL Pipeline (Extract, Transform, Load)

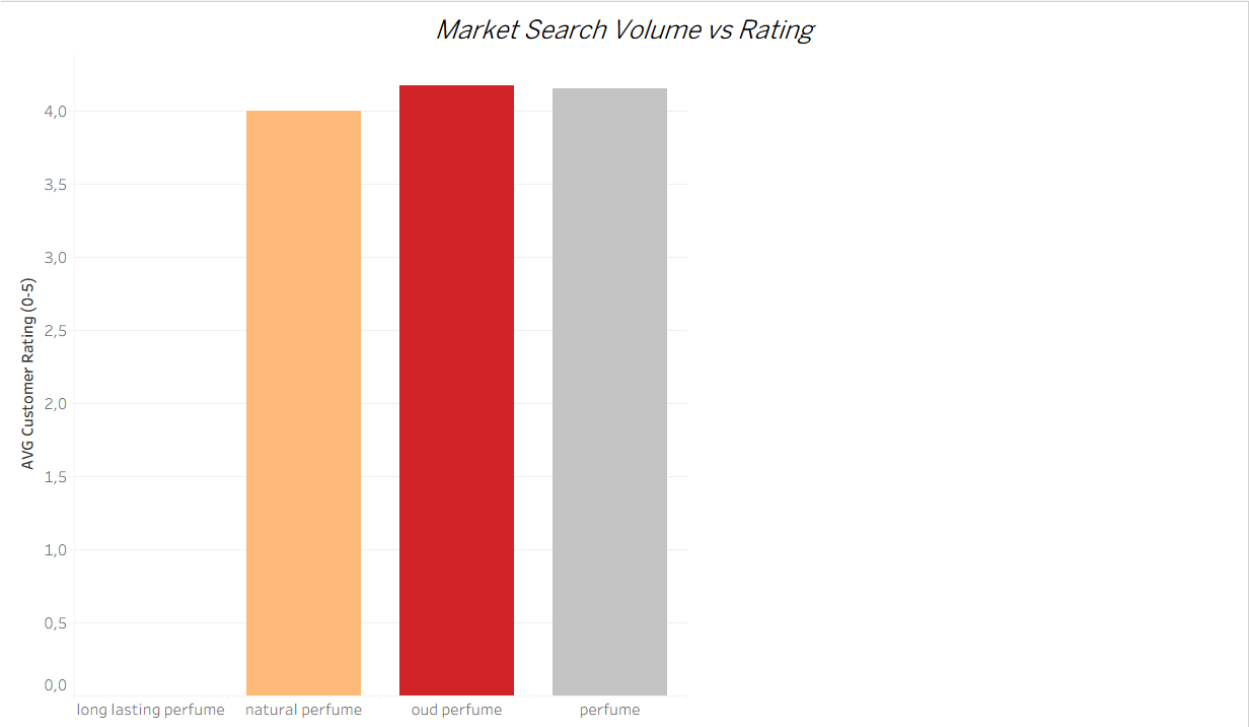
Raw data from disparate sources required significant cleaning before storage. I utilized Pandas for this transformation.

- **Normalization:** Convert various rating scales (0-10) into a standardized 1-5 scale.
- **Standardization:** Merged synonymous ingredients (e.g., "Oudh", "Aoud", "Agarwood") into a single "Oud" category.

Imputation: Filled missing values in the 'Main Accords' column using the mode of the Brand's portfolio.

Exploratory Visualization

Initial analysis revealed a key insight: while 'Natural' perfumes have higher volume, 'Oud' perfumes have significantly higher average ratings.



Database type selection

I selected **MySQL (Relational Database)** for the core storage.

- **Why Relational?** The data has a strict schema: A *Brand* has many *Perfumes*. A Perfume belongs to one *Category*. This structured relationship requires ACID compliance to ensure data integrity for the API.

- **Why MySQL?** Our querying needs (complex JOINs between Products and Trends) are far more efficient in SQL.

Entities

To support the analysis, I designed a database schema with 4 core entities connected via shared keys.

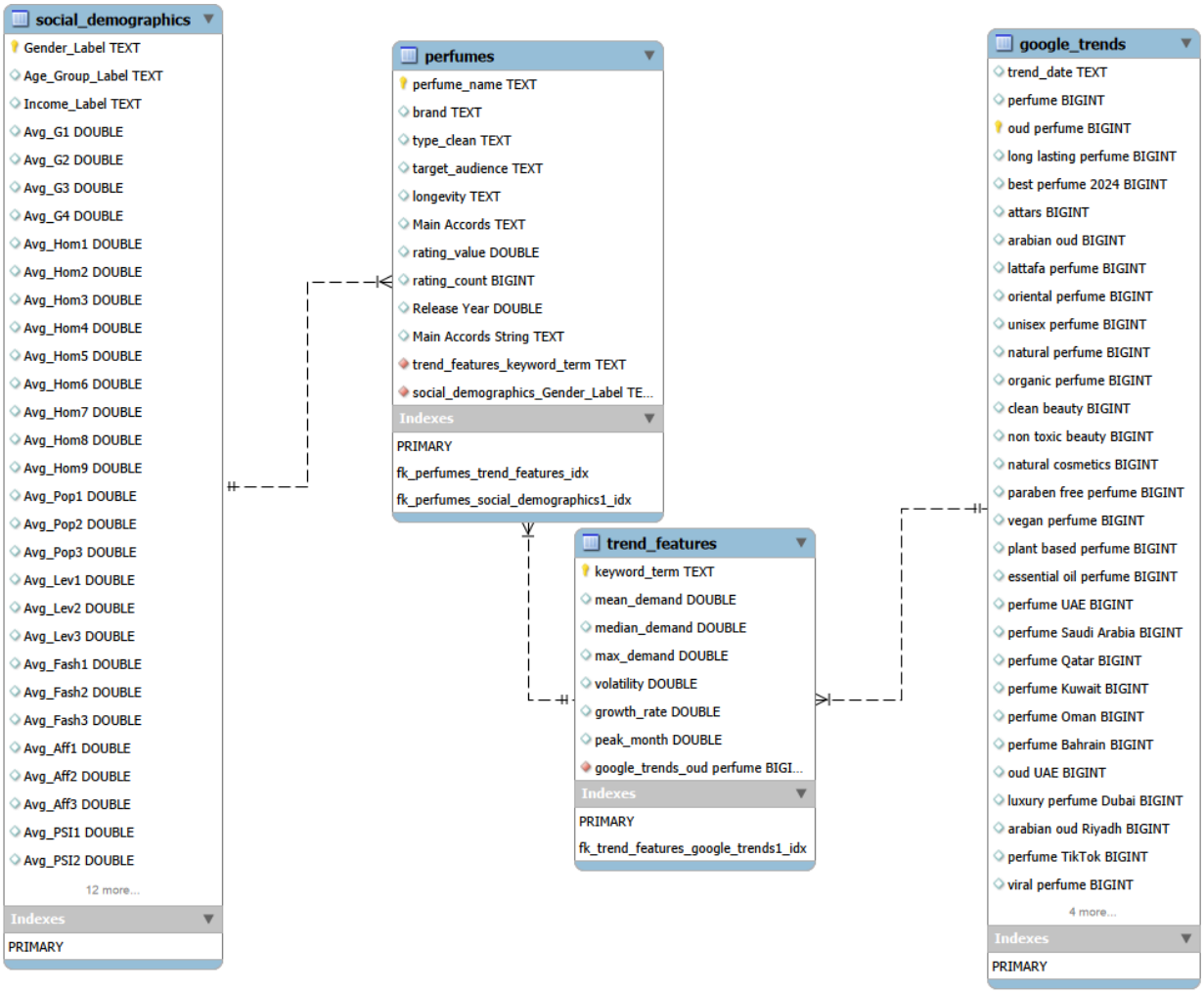
Entities:

1. **Perfumes:** The central table containing product metadata.
    - *PK:* id (auto-generated)
    - *Attributes:* perfume_name, brand, rating_value, main_accords.
2. **Trend_Features:** Market demand metrics.
    - *Join Key:* keyword_term
    - *Attributes:* mean_demand, growth_rate.
3. **Social_Demographics:** Customer profiles and sentiment.
    - *Join Key:* Gender_Label (maps to perfumes.target_audience)
    - *Attributes:* Age_Group, Income_Label.
4. **Google_Trends:** Raw time-series search data.
    - *Join Key:* keyword (maps to Trend_Features.keyword_term)

Relationships (Logical Model):

- **Perfumes (1,n) --- (0,1) Trend_Features:**
    - *Logic:* Each perfume name (e.g., "Oud Wood") maps to a specific search keyword in the trends table. Not all perfumes have trending keywords (0,1), but popular ones do.
- **Perfumes (1,n) --- (1,1) Social_Demographics:**
    - *Logic:* Perfumes target a specific audience (Male/Female/Unisex). This links to the demographic profiles in the social table.
- **Trend_Features (1,1) --- (1,n) Google_Trends:**
    - *Logic:* A single trend keyword (e.g., "Oud") has many daily search records (Time Series) in the raw Google Trends table.

ERD


social_demographics
Gender_Label TEXT
Age_Group_Label TEXT
Income_Label TEXT
Avg_G1 DOUBLE
Avg_G2 DOUBLE
Avg_G3 DOUBLE
Avg_G4 DOUBLE
Avg_Hom1 DOUBLE
Avg_Hom2 DOUBLE
Avg_Hom3 DOUBLE
Avg_Hom4 DOUBLE
Avg_Hom5 DOUBLE
Avg_Hom6 DOUBLE
Avg_Hom7 DOUBLE
Avg_Hom8 DOUBLE
Avg_Hom9 DOUBLE
Avg_Pop1 DOUBLE
Avg_Pop2 DOUBLE
Avg_Pop3 DOUBLE
Avg_Lev1 DOUBLE
Avg_Lev2 DOUBLE
Avg_Lev3 DOUBLE
Avg_Fash1 DOUBLE
Avg_Fash2 DOUBLE
Avg_Fash3 DOUBLE
Avg_Aff1 DOUBLE
Avg_Aff2 DOUBLE
Avg_Aff3 DOUBLE
Avg_PSI1 DOUBLE
Avg_PSI2 DOUBLE
12 more...
Indexes
PRIMARY
perfumes
perfume_name TEXT
brand TEXT
type_clean TEXT
target_audience TEXT
longevity TEXT
Main Accords TEXT
rating_value DOUBLE
rating_count BIGINT
Release Year DOUBLE
Main Accords String TEXT
trend_features_keyword_term TEXT
social_demographics_Gender_Label TE...
Indexes
PRIMARY
fk_perfumes_trend_features_idx
fk_perfumes_social_demographics1_idx
trend_features
keyword_term TEXT
mean_demand DOUBLE
median_demand DOUBLE
max_demand DOUBLE
volatility DOUBLE
growth_rate DOUBLE
peak_month DOUBLE
google_trends_oud perfume BIGI...
Indexes
PRIMARY
fk_trend_features_google_trends1_idx
google_trends
trend_date TEXT
perfume BIGINT
oud perfume BIGINT
long lasting perfume BIGINT
best perfume 2024 BIGINT
attars BIGINT
arabian oud BIGINT
lattafa perfume BIGINT
oriental perfume BIGINT
unisex perfume BIGINT
natural perfume BIGINT
organic perfume BIGINT
clean beauty BIGINT
non toxic beauty BIGINT
natural cosmetics BIGINT
paraben free perfume BIGINT
vegan perfume BIGINT
plant based perfume BIGINT
essential oil perfume BIGINT
perfume UAE BIGINT
perfume Saudi Arabia BIGINT
perfume Qatar BIGINT
perfume Kuwait BIGINT
perfume Oman BIGINT
perfume Bahrain BIGINT
oud UAE BIGINT
luxury perfume Dubai BIGINT
arabian oud Riyadh BIGINT
perfume TikTok BIGINT
viral perfume BIGINT
4 more...
Indexes
PRIMARY

SQL Analysis

To extract actionable insights, I developed 5 key SQL scripts.

Script 1: *Volume Leaders (Affordable Market)*

Goal: Identify brands dominating market share.

```sql
SELECT
    brand,
    COUNT(*) AS oud_perfume_count
FROM perfumes
WHERE LOWER(`Main Accords`) LIKE '%oud%'
   OR LOWER(perfume_name) LIKE '%oud%'
GROUP BY brand
ORDER BY oud_perfume_count DESC
LIMIT 5;
```

Script 2: *Quality Leaders (Luxury Market)*

Goal: Identify highest-rated products with significant validation.

```sql
SELECT
    perfume_name,
    brand,
    rating_value as Rating,
    rating_count as Votes
FROM perfumes
WHERE (LOWER(`Main Accords`) LIKE '%oud%' OR LOWER(perfume_name) LIKE '%oud%')
  AND rating_count > 50
ORDER BY rating_value DESC
LIMIT 5;
```

Script 3: *Category Performance Comparison*

Goal: Validate the "Oud vs Natural" hypothesis.

```sql
SELECT
    Trend_Category,
    COUNT(*) as Product_Volume,
    ROUND(AVG(rating_value), 2) as Avg_Satisfaction_Score
FROM v_perfume_market_trends
GROUP BY Trend_Category
ORDER BY Avg_Satisfaction_Score DESC;

```

Script 4: *Trend Growth Analysis*

Goal: Identify the fastest-rising search terms.

```sql
SELECT
    keyword_term,
    growth_rate,
    mean_demand
FROM trend_features
ORDER BY growth_rate DESC
LIMIT 5;
```

Script 5: *Social Sentiment Correlation*

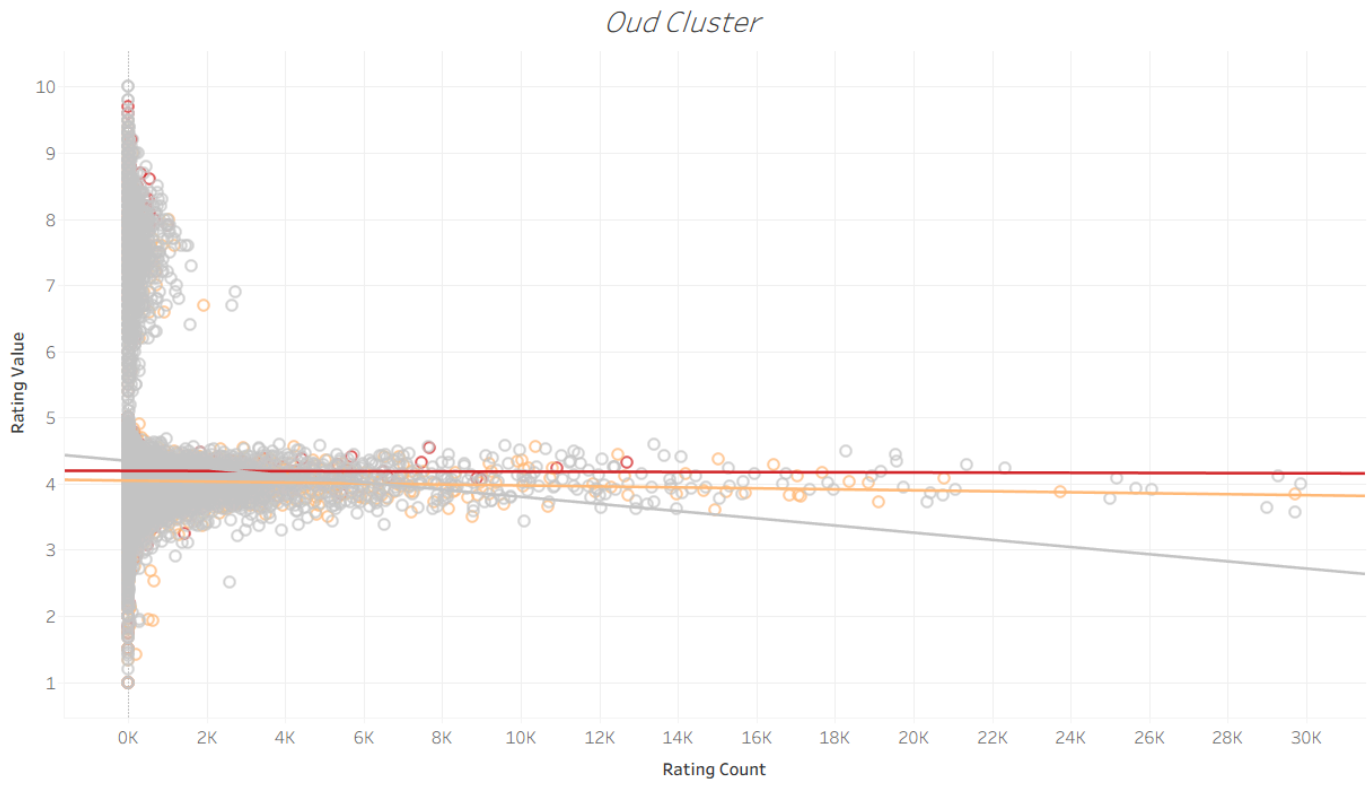
Goal: Correlate search demand with product quality.

```sql
SELECT
    v.Trend_Category,
    ROUND(AVG(v.rating_value), 2) as avg_product_rating,
    ROUND(AVG(t.mean_demand), 2) as avg_search_volume
FROM v_perfume_market_trends v
JOIN trend_features t ON v.Trend_Category = t.keyword_term
GROUP BY v.Trend_Category
ORDER BY avg_search_volume DESC;
```

Key Insights from SQL Analysis

Our queries revealed critical market dynamics:

- Oud perfumes achieve higher ratings (4.17) than natural (4.0) or general perfumes (4.15), despite representing only 11% of the market.
- Regional trends are decisive: 'Oud UAE' and 'Perfume Qatar' show the highest search demand and growth, signaling untapped geographic opportunities.
- Long-lasting perfumes are a white space: only 1 product exists, yet search growth is 64%, indicating massive potential.
- Luxury brands (Xerjoff, Amouage) dominate the top-rated perfumes, proving that quality and exclusivity drive success.



Rating Value vs. Rating Count for Oud, Natural, and General Perfumes. Oud perfumes not only receive higher average ratings but also exhibit a stronger correlation between rating count and rating value, indicating consistent quality recognition.

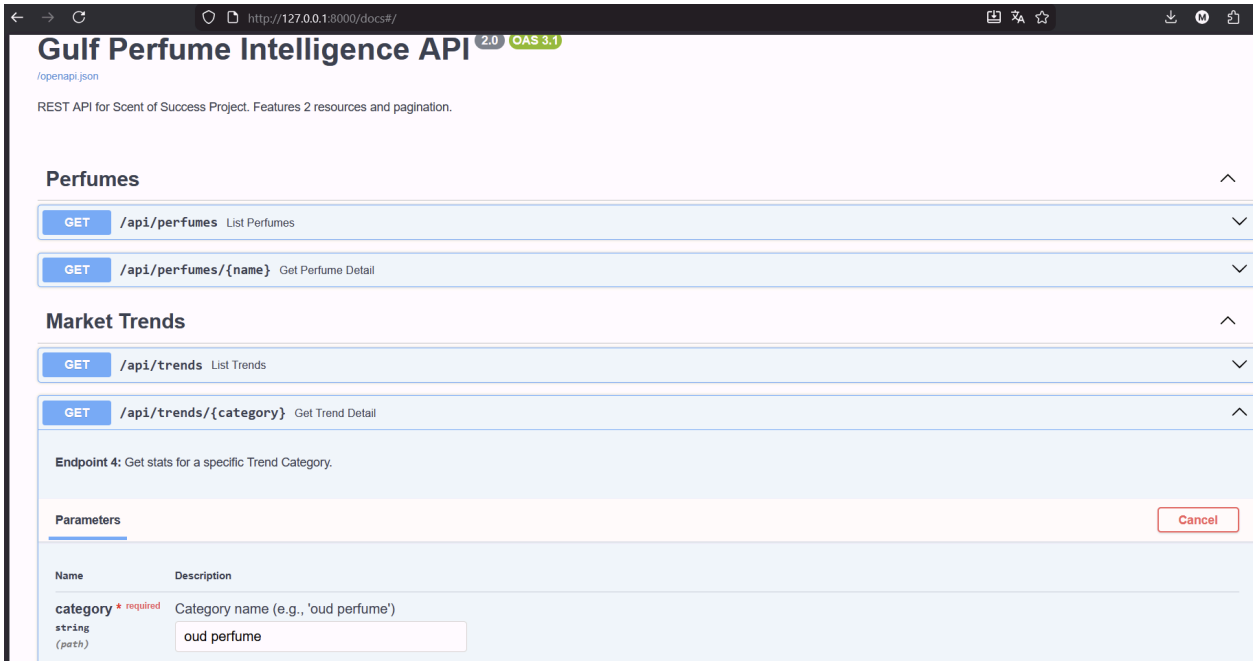
The combination of SQL analysis and visualizations confirms that oud perfumes represent a high-value, high-satisfaction niche. For brands, this means prioritizing oud-based formulations and targeting regional demand hotspots like the UAE and Qatar.

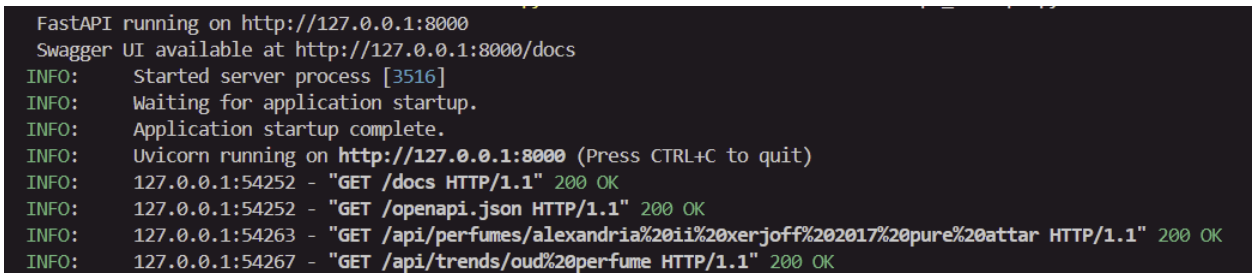
API

I developed a REST API. I implemented endpoints covering **2 Resources** and **4 Endpoints** with pagination.

Architecture:

- **Resource 1: /api/perfumes** (List & Detail endpoints with pagination).
- **Resource 2: /api/trends** (Market intelligence endpoints).





GDPR

Privacy by Design Strategy:

The "Social Demographics" dataset initially contained sensitive User IDs and IP addresses. To ensure GDPR compliance:

1. **Anonymization:** I implemented a pre-ingestion filter in Python that drops *IP_Address* and *User_ID* columns.
2. **Aggregation:** Data is stored only in aggregated cohorts (e.g., "Age Group: 18-24") rather than individual records.

This ensures the system never stores Personally Identifiable Information (PII).

CONCLUSION

The project demonstrates a complete, production-ready data lifecycle: from complex multi-source collection (Scraping/BigData), through cleaning and relational storage, to secure and documented exposition via API.

“Scent of Success” isn’t just a pipeline, it’s a proof of concept for data-driven decision-making in fast-moving markets. With the foundation in place, the next step is to integrate generative AI, automating insights and empowering brands to act on trends before they become mainstream.

This is the future of market intelligence.