Name : Yuvraj Singh Deol
Final Project for Minor in AI from IIT Ropar
Student ID : iitrpr_ai_25010735
Project : AI Experience Market Analyser and Planner

Problem Statement

There has been a rising trend in todays Gen Z and Upper Middle Classes of the world and in India which is that people have shifted from spending money on materialistic things like cars , bags etc and have moved on to "Experiences" Like Concerts , Stand-ups and Workshops.
But as organisers , they work on guesses this reliance on guesswork leads to three major issues:

Poor Turnout: Launching events during low interest periods or in markets with low money spending power

Pricing Inefficiency: Undervaluing tickets in high demand cities or overpricing in price sensitive markets.

Wasted Marketing Budget: Spending resources on events that were not going to work due to seasonality or location

The objective of this project is to build an **AI Experience Market Analyser and Planner**. This tool analyses historical demand signals (Google Trends) and seasonality patterns to provide data-backed recommendations on whether to add a show, pause it, or adjust pricing.

Methodology

The project follows a standard Data Science pipeline involving ETL (Extract, Transform, Load), Feature Engineering, and a Heuristic Decision Engine.

Step 0 (Data Collection)

I Downloaded google trends data of 35 keywords from google trends of india for the past year using comet web browser

Generated seasonality and City multiplier scores using ai and census data of general trends and the amount of money spending power of cities respectively

Step 1 (Data Ingestion & Preprocessing)

The system ingests data from 35 distinct CSV files representing Google Trends "multiTimeline" data for keywords like "Concert," "Standup," and "Workshops." It also ingests a "Seasonality" dataset that provides month-wise weights (e.g., higher weights for festival months). The data is cleaned, date-parsed, and converted from a wide format to a long format to allow for time-series analysis.

Step (2) Feature Engineering: The Demand Index

To quantify market viability, a composite score called the Demand_Index is made.

This index is a weighted average that balances raw search interest with seasonal validity:

Demand Index = (0.7 * Trend_Score) + (0.3 * Season_Score)

This ensures that a high search volume is validated by a favourable time of year.

City-Based Economic Adjustments A heuristic "City Multiplier" is applied to account for the varying purchasing power across Indian cities. For example, Tier 1 cities like Mumbai are assigned a multiplier of 1.30 (indicating higher spending capacity), while smaller markets like Indore are adjusted to 0.95. This allows the same base event to generate different pricing recommendations based on location.

4. Decision Logic The intelligence of the system compares the calculated market demand against the user's proposed ticket price. It uses the following logic to generate recommendations:

- **Increase Price:** If demand is high and the user's price is significantly lower (>15%) than the recommended value.
- **Decrease Price:** If the user's price exceeds the market's calculated willingness to pay.
- **Remove/Pause:** If the Demand_Index falls below a critical threshold (e.g., <25), indicating that holding the event is too risky.

Tools & Technologies

The project was built using a Python based stack designed for rapid prototyping and interactive data visualisation:

- **Programming Language:** Python 3 (Core logic and data processing).
- **Data Manipulation: Pandas** library for merging datasets, handling missing values (errors='coerce'), and vectoring the demand calculations.
- **User Interface (UI): Streamlit** framework to create the interactive dashboard. It allows users to select Show Type, City, Month, and Price with drop downs and sliders.
- **Deployment/Tunneling: Cloudflared** (Cloudflare Tunnel) was used to expose the local Streamlit server running in the Colab notebook to a public URL for remote access.
- **Data Source:** Google Trends (Interest over time CSVs) and custom Seasonality datasets hosted on GitHub.

Results & Conclusion

Results The final output is a fully functional dashboard titled "Show Forecast Dashboard."

1. **Dynamic Recommendations:** When a user inputs a scenario (e.g., "Standup Comedy in Mumbai in December at ₹2000"), the system successfully calculates a "City Adjusted Demand" and provides a specific recommendation, such as "*KEEP: Your price is close to recommended ₹2100*" or "*DECREASE PRICE.*"

2. **Planning Utility:** The system includes a "Plan Table" feature, allowing organisers to add valid events to a list and download a CSV plan

3. **Robustness:** The code successfully handles edge cases, such as missing city data (defaulting to a 1.0 multiplier) or null trend values, ensuring the dashboard does not crash during use.