Project Report

KSA Brand Licensing Suitability Analysis Tool.

1. Project Goal & Intention

The primary objective of this project was to develop a data-driven tool to analyze and quantify the licensing suitability of brands within the Kingdom of Saudi Arabia (KSA). The intention was to move beyond simple popularity metrics and provide a multi-dimensional analysis for strategic business decisions.

The tool is designed to answer four key questions for a potential licensor or investor:

1. **Hype (Demand):** How much public conversation does the brand generate? Is that conversation positive or negative?
2. **Market Presence (Supply):** How saturated is the local e-commerce market (Amazon.sa) with this brand's products?
3. **Perceived Quality:** What is the average consumer rating of existing products?
4. **Product Popularity:** Are existing products being frequently purchased and reviewed?

The final deliverable is a functional Graphical User Interface (GUI) application that consolidates these metrics into a single **"Suitability Score"** and presents a comprehensive report, including a comparative radar chart visualization.

---

2. Phase 1: Data Foundation & Collection (ETL)

This phase focused on identifying target brands and acquiring the necessary raw data.

2.1. Target Brand Curation

Based on conference participant lists [cite: image_571f1b.png, image_571ec4.png, image_571e81.png, image_571bbc.png] and market research, we curated a target list of **65 KSA-relevant, merchandise-focused brands**. This list was categorized to ensure relevance:

- **Food & Beverage:** Almarai, Saudia Dairy (SADAFCO), Herfy, Kudu, Al Rabie
- **Fashion, Health & Beauty:** Lazurde, Nahdi, Jarir Bookstore, Arabian Oud, Abdul Samad Al Qurashi, Mikyajy

- **Sports & Lifestyle:** Al-Hilal, Al-Nassr, Al-Ittihad, Fanatics, Fitness Time
- **Cultural & Niche:** KSA Anime, KSA One Piece, Al Romansiah, Sleysla, Camel Step
- **Major Retailers:** SACO, eXtra, Mall of Arabia, Riyadh Park Mall

Corporate-only entities (e.g., PIF, Saudi Aramco) were intentionally excluded from the e-commerce analysis to maintain a focus on merchandise-licensing.

2.2. Data Sources & Collection Tools

- **Social Hype Data:** Twitter (X)
- **Market Reality Data:** Amazon.sa
- **Scraping Tool: Apify** platform, accessed via the apify-client library in Python.

2.3. Data Collection: Challenges & Solutions

This was the most complex phase, defined by significant technical challenges.

- **Challenge 1: Initial Scrapers Failed**
  - Our first approach using local Python libraries (ntscraper, requests, BeautifulSoup) failed entirely due to aggressive anti-bot measures, IP blocking, and CAPTCHAs from all target sites.
- **Solution 1: Pivot to Apify**
  - We pivoted to a professional-grade "Seed Scrape" strategy, using Apify's robust infrastructure and proxy networks. This allowed us to bypass the blocking mechanisms.
- **Challenge 2: Apify Credit Limit**
  - While scraping the expanded 65-brand list, we exhausted the initial $5 free credit limit [cite: User logs showing "Monthly usage hard limit exceeded"].
- **Solution 2: Resume Scripts**
  - A new Apify account with fresh credits was utilized. We created new, targeted "resume" scripts (1_scrape_hype_resume.py, 2_scrape_ecommerce_resume.py) that scraped *only* the brands missed by the first run, successfully completing the baseline dataset.
- **Challenge 3: Twitter Data Corruption (The Critical Fix)**

    - Initial analysis showed that our tweets table (5,852 rows) contained **no tweet text, no engagement counts, and no valid dates**.
- **Solution 3: Debugging & Re-Scraping**
    - By analyzing the raw JSON output from the xtdata/twitter-x-scraper actor [cite: dataset_twitter-x-scraper_2025-10-23_23-17-05-931.json], we identified a critical error in our script: we were saving the wrong field names.
    - **Fixed Fields:**
        - text ➔ full_text
        - createdAt ➔ created_at
        - likeCount ➔ favorite_count
        - userName ➔ author['screen_name']
    - We executed a corrected 1_scrape_hype.py (v2), which first **deleted the 5,852 bad rows** and then re-ran the scrape for all 65 brands, resulting in a new, valid dataset of **17,386 tweets**.
- **Challenge 4: Amazon Data Parsing**
    - Initial analysis showed Saved 0 new Amazon products, indicating a parsing failure.
- **Solution 4: Debugging & Re-Scraping**
    - By analyzing the raw JSON from the junglee/Amazon-crawler actor [cite: dataset_Amazon-crawler_2025-10-26_01-32-16-487.json], we identified and corrected the field names in our 2_scrape_ecommerce_apify.py script.
    - **Fixed Fields:**
        - URL ➔ Built from asin (e.g., https://www.amazon.sa/dp/ASIN)
        - Price ➔ price['value']
        - Rating ➔ stars
        - Review Count ➔ reviewsCount
    - A re-run of this script successfully collected **1,020 product listings**.
- **Challenge 5: Counterfeit Risk Data (Amazon Reviews)**

    - A key goal was to analyze review text for counterfeit risk.

- **Solution 5: Acknowledgment & Pivot**
    - We systematically tested multiple Apify actors (web_wanderer/amazon-reviews-extractor, epctex/amazon-reviews-scraper, etc.). All of them **failed** to reliably scrape the amazon.sa region due to internal actor bugs (TypeError: startUrl.includes is not a function) or specific input validation failures for that domain.
    - We made the strategic decision to **skip this data source for v2** and proceed without a Counterfeit Risk Score, prioritizing a functional end-to-end product.

---

3. Phase 3: Analysis & Feature Engineering (EDA)

This phase was conducted in a Jupyter Notebook (EDA_and_Modeling_v2.ipynb) using **Pandas**.

1. **Data Cleaning:**
    - Loaded the 17,386 valid tweets and 1,020 valid products from SQLite.
    - Successfully parsed the created_at column to datetime objects using the correct format string (%a %b %d %H:%M:%S +0000 %Y).
    - Created a cleaned_content column by removing URLs and special characters from tweet text.
    - Filled missing avg_rating and num_reviews in the product data with 0.
2. **Feature Engineering (Calculated Metrics):**
    - **Tweet Volume:** Total COUNT() of tweets per brand.
    - **Total Engagement:** SUM(like_count + retweet_count + ...) per brand.
    - **Average Tweet Sentiment:** Used **VADER** on the cleaned_content of all 17k+ tweets to generate a compound score (-1 to +1) and averaged this per brand.
    - **Topic Modeling (Brand DNA):** Used **NLTK** stopwords and **Scikit-learn**'s TfidfVectorizer and LatentDirichletAllocation (LDA) on the cleaned_content to identify the Top 5 key themes for each brand (e.g., Al-Nassr: "Ronaldo", "league"; Jarir: "offers", "books").
    - **Market Saturation:** Total COUNT() of Amazon products per brand (capped at 25).

    - **Perceived Quality:** AVG(stars) per brand (0-5 scale).
    - **Product Popularity:** AVG(reviewsCount) per brand.

---

4. Phase 4: The "Model" - Brand Suitability Score

To create a single, actionable metric, we developed a weighted scoring system. All features were first normalized to a 0-100 scale (e.g., tweet_volume normalized by max volume, avg_perceived_quality normalized from its 0-5 scale, market_saturation inversely normalized).

Suitability Score (0-100) =

- (Normalized Hype Volume * 30%)
- (Normalized Tweet Sentiment * 20%)
- (Normalized Product Quality * 25%)
- (Normalized Product Popularity * 15%)
- (Normalized *Low* Saturation * 10%)

This score and its components were saved to data/brand_metrics_final_v2.csv.

---

5. Phase 5: Final Application & Visualization (consultant_tool.py)

The final deliverable is a standalone GUI application built with **Tkinter** and **Matplotlib**.

- **Functionality:**
    1. **Data Pre-Loading:** On startup, the app loads all pre-processed metrics from brand_metrics_final_v2.csv and raw product data from licensing_data.db.
    2. **Autocomplete Search:** Features a ttk.Combobox that allows the user to either type a brand name (with real-time filtering) or select from the full dropdown list, ensuring no "multiple matches" errors.
    3. **Instant Report Generation:** Clicking "Generate Report" queries the pre-loaded DataFrames.
- **Report Output:** The GUI presents a comprehensive, two-panel report:
    1. **Text Panel (Left):**
        - **Overall Score:** The final Suitability Score (e.g., "75.1 / 100").

- **Recommendation:** A qualitative summary (e.g., "HIGH POTENTIAL").
- **Metrics Breakdown:** A formatted table showing the raw value, normalized 0-100 score, and percentile rank for each metric (Hype, Sentiment, Quality, etc.).
- **Top 5 Products:** A list of the brand's top-rated Amazon.sa products.

2. **Visualization Panel (Right):**
   - **Radar Chart:** A dynamic Matplotlib plot comparing the brand's 5 key normalized metrics against the average of all other brands in the dataset, providing an instant visual profile [cite: image_064fc0.png].

---

6. Future Expansion Opportunities

This v2 tool is a powerful prototype. The pipeline is designed for significant expansion:

1. **Fully Automated ETL Pipeline:** Create the run_pipeline.py master script. Build new, free, local scrapers (requests, snscrape) to run on a schedule (e.g., GitHub Actions), **append** new data to the SQLite DB, and re-generate the final metrics CSV. This makes the tool "real-time" and free to maintain.
2. **Implement Counterfeit Risk (Priority):** Re-attempt the Amazon review scrape (Phase 3c). This may require a different scraping service (e.g., ScrapingBee) or building a more robust local scraper with Selenium/Playwright to finally get the review text.
3. **Integrate LLMs (Gemini API):**
   - **Advanced Sentiment/Topics:** Replace VADER and LDA with calls to a generative model. This would provide far more nuanced sentiment scores (especially for Arabic) and human-readable topic summaries.
   - **Review Summarization (Post-Scrape):** Once review text is captured, an LLM could summarize thousands of reviews into "Top 3 Pros" and "Top 3 Cons," which would be displayed in the GUI.
   - **Automated Insights:** An LLM could be fed the final metrics (e.g., "Hype=90, Quality=20, Risk=High") to generate a complete, qualitative paragraph explaining *why* the brand is a good or bad opportunity.

4. **Specialized Corporate Models:** Build separate, parallel models for the non-merchandise brands (PIF, Aramco) that scrape financial news (e.g., via Google Search API) and measure corporate sentiment or investment buzz instead of Amazon product data.
5. **Expand Data Sources:** Integrate data from TikTok (via relevant scrapers) and Instagram (via official APIs or scrapers) to create a more holistic "Hype" score.