Perfume Recommendation Model Proposal

Business Understanding

Introduction
The perfume industry faces increasing competition, with many products available in various scents, brands, and price ranges. A personalized perfume recommendation model can enhance customer engagement and satisfaction by helping users discover scents suited to their preferences. This model will leverage perfume data and user preferences to offer tailored recommendations, ultimately increasing purchase likelihood and brand loyalty.

Problem Statement
Customers often find it challenging to select perfumes that match their preferences, when shopping online or at the nearest perfume store. While shopping online, there is minimal opportunity to sample scents directly and clients will only rely on reviews, fragrance descriptions, and recommendations. This project aims to develop a machine learning-based recommendation system that suggests perfumes based on users' preferred fragrance notes, categories (e.g., unisex, men's, women's), and price ranges, improving the customer shopping experience.

Domain
This project lies in the intersection of e-commerce, fragrance retail, and recommendation systems. It involves using machine learning to deliver a personalized shopping experience that aligns with the customer's scent profile and budget.

Target Audience
The target audience includes:
- Fragrance Enthusiasts who enjoy discovering new scents.
- E-commerce Retailers looking to enhance their online shopping experience with personalized recommendations.
- Fragrance Brands that want to improve product discoverability and increase sales.

Impact
A successful perfume recommendation model will improve user satisfaction, engagement, and conversion rates. By helping customers discover perfumes they are likely to enjoy with ease, the system can increase the likelihood of repeat purchases and build brand loyalty.

Pre-existing Work
Recommendation systems are well-studied in e-commerce and media domains (e.g., Amazon, Netflix), with both content-based and collaborative filtering approaches commonly used. This project builds on existing content-based recommendation technique but tailors them for the fragrance industry by incorporating perfume-specific attributes like fragrance notes, categories and prices.

Success Criteria
***- Recommendation Quality: Measured by accuracy metrics such as precision, recall, and F1-score.
- User Satisfaction: A measure of how often recommended perfumes match user preferences, possibly obtained through surveys or feedback loops.
- Engagement Metrics: Click-through and conversion rates for recommended products.

- Revenue Impact: Increase in sales or average order value attributable to recommendations.

Objectives
1. Develop a recommendation engine that can suggest perfumes based on fragrance notes, categories, and price.
2. Implement a user-friendly interface to allow customers to input preferences (e.g., fragrance notes, budget) and receive relevant recommendations.
3. Increase user engagement by 15% within the first quarter post-deployment by improving product discovery and relevance.

Data Understanding

Data Source
The dataset was scraped using `cierra.ipynb`, which pulls information from online fragrance retailer, i.e, https://cierraperfumes.com/. It includes information about each perfume's name, notes, category (e.g., unisex, men's, women's), price, brand, and possibly customer ratings or reviews.

Data Features
Key data features include:

1. Category: The gender target, i.e, women, men or unisex
2. Title: The name of the perfume
3. Price: The price of the perfume in Ksh.
4. Link: The specific location on Cierra website
5. Image: The location of the perfume packaging image
6. Description: A synopsis of the perfume
7. Top: The initial scent, lasting 5-15 minutes. Usually light, fresh, or citrusy to capture attention.
8. Middle: The cores of the fragrance, emerging after the top notes dissipate, lasting 30 minutes to an hour. Often floral or spicy, these are the perfume's signature
9. Base: The final lingering notes, which emerge after the middle notes fade and can last several hours. They are typically rich and include woods, musk, and resins.

Data Preparation
1. Data Cleaning: Remove duplicates, handle missing values, and standardize textual data (e.g., consistent formatting for notes and brands).
2. Feature Engineering:
  - One-Hot Encoding for categorical variables like fragrance categories.
  - TF-IDF Vectorization for text-based attributes such as fragrance notes.
  - Normalization of numerical features like price to make them comparable.
3. Data Splitting: Split data into training and testing sets to ensure the model's generalization capability.

Modeling
1. Content-Based Filtering:
  - Use cosine similarity on fragrance notes and categories to recommend perfumes similar to those a user likes.

- Apply TF-IDF to convert perfume notes into feature vectors.

**Evaluation

To measure the effectiveness of the recommendation model, the following metrics will be used:
1. Gender Relevance (1.0000): This indicates a full emphasis on gender-appropriate fragrances. The recommendations prioritize options specifically for women, which ensures the suggestions are aligned with audience.

2. Note Relevance (1.0000): This indicates a low emphasis on the fragrance notes in the product recommendations. It suggests that the specific scent profile may not be a primary factor in the selection process.

3. Price Relevance (1.0000): This shows a strong focus on prices, suggesting that the recommendations take into account the price range that is likely to resonate with the intended consumer.

Deployment

1. API Development: Develop a REST API using Flask or FastAPI to provide recommendations in real-time.
2. User Interface: Create a web interface or integrate into an existing e-commerce platform, where users can enter their preferences (e.g., favorite notes, budget).
3. Monitoring and Feedback: Track recommendation accuracy and relevance through user feedback and engagement metrics. Implement periodic model retraining with new data to adapt to changing user preferences.

Tools/Methodologies

- Data Collection: Scraping tools (BeautifulSoup, Selenium) in Python.
- Data Processing and Analysis: Pandas, NumPy, and scikit-learn.
- Modeling: Scikit-learn for content-based filtering

- Deployment: Flask or FastAPI for API, with possible integration into a web platform.
- Evaluation: Custom metric functions in Python, utilizing metrics from scikit-learn.