Deliverable 1 — From Pitch to Prototype: Foundation and Design Blueprint

Perfume Style Categorization and Recommendation

Name: Srinija Jonnavithula
Applied Machine Learning II (EEE 6778)

GitHub Repository:
github.com/sjonnavithula09/Perfume-Style-Categorization-and-Recommendation

a. Problem Context and Project Summary

Selecting the right perfume is subjective and context-dependent. Fragrances differ by season, climate, and personal style; what feels fresh in summer may seem weak in winter, and gendered marketing rarely captures true preferences. Existing discovery relies on static tags or marketing keywords, lacking personalization and explainability.

This project proposes an **AI-driven perfume style categorization and recommendation system** where users select a style—*Summer, Winter, Fall, Women, Men, or Unisex*—and receive ranked perfume suggestions with brief, transparent explanations. The goal is to make fragrance selection data-driven, interpretable, and interactive.

b. Dataset

Source: Curated public fragrance datasets (e.g., Kaggle/open fragrance note collections) plus limited manual curation.
Type/size: Structured + text (≈ 1,000–1,500 entries).
Fields: `id, brand, name, year, notes_top, notes_heart, notes_base, accords, description, gender_tag (opt.)`.
Format/access: CSV → Pandas; stored in `data/` with citation in README.
Preprocessing: Normalize note names (e.g., "Calabrian bergamot" → "bergamot"), deduplicate, handle missing text, tokenize descriptions, build note co-occurrence matrix.

c. Planned Architecture

Flow: Data → Transformer Encoder → Note Graph (GNN) → Fusion → Classifier/Retriever → Streamlit UI.

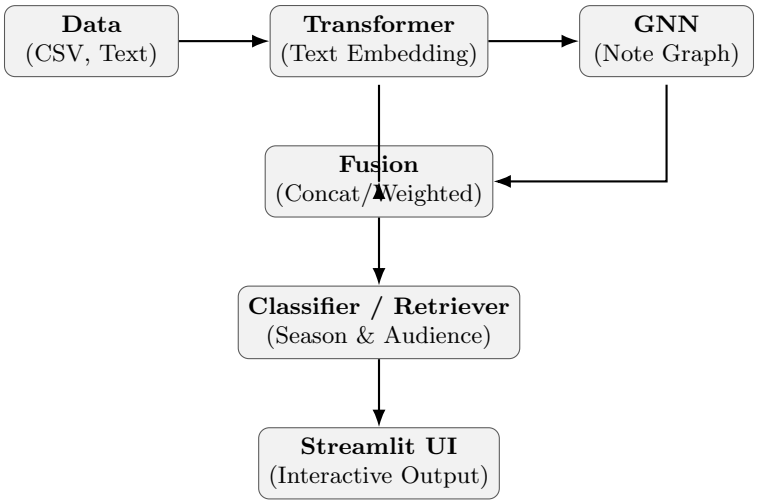

Figure 1. Compact architecture: data through models to interface.

Components.

- **Transformer Encoder:** SBERT/MiniLM embeddings of description + flattened note pyramid.
- **GNN over Note Graph:** GraphSAGE/GCN on note co-occurrence; aggregate a perfume's note vectors.
- **Fusion:** LayerNorm($W_t z_{text} \oplus W_g z_{notes}$).
- **Heads:** (i) Multi-label classifier for Season/Audience; (ii) cosine-similarity retriever with MMR for diversity.
- **Explainability:** Top contributing notes and salient phrases aligned to the chosen style.

Frameworks: PyTorch, PyTorch Geometric, Transformers; Matplotlib/Plotly for visuals.

d. User Interface Plan

Input: Dropdowns for Season and Audience (e.g., Summer + Unisex).
Process: Retrieve top-k perfumes via cosine similarity in fused space.
Output: Ranked cards displaying brand, name, notes/accords, and a short "why it matches" rationale. Optional mini-graph visualization highlights influential notes.

e. Innovation and Anticipated Challenges

Innovation.

- Hybrid **Transformer + GNN** representation for perfume text/notes.
- **Weak supervision** for Season/Audience labels using heuristics + text prompts, with small gold calibration.
- **Explainable** retrieval with ingredient-level attribution.

Challenges and Mitigation.

Challenge	Mitigation Strategy
Sparse or noisy style labels	Use heuristic note weights and prompt similarity; combine multiple weak signals; audit 200+ samples to calibrate thresholds.
Limited dataset volume	Employ pre-trained encoders, shallow GNN layers, and lightweight data augmentation; cache embeddings for reuse.
Latency or model complexity	Precompute perfume embeddings; limit GNN depth to 1–2 layers; apply cosine retriever with Maximal Marginal Relevance (MMR) for efficiency.

f. Implementation Timeline

Week	Focus	Expected Outcome
Oct 20–26	Data collection/cleaning, EDA	Working data loader; note normalization; co-occurrence matrix.
Oct 27–Nov 2	Baseline retrieval	Heuristic + text-prompt scorer; simple top-k demo.
Nov 3–9	Transformer classifier	Season/Audience probabilities; initial metrics.
Nov 10–16	Note graph + GNN fusion	Fused embeddings; improved categorization.
Nov 17–23	Streamlit prototype	Interactive top-k list with explanations.
Nov 24–30	Eval + explainability	Precision@k/nDCG; note attributions; polish.
Dec 1–11	Final polish	Poster, report, and stable demo.

g. Responsible AI Reflection

- **Fairness:** Treat Women/Men/Unisex as stylistic clusters; surface neutral "Unisex" by default when ambiguous.
- **Transparency:** Show note-level evidence and brief textual rationales for each recommendation.
- **Environment:** Favor small encoders; shallow GNN; cache embeddings to reduce compute.