**Title of the Project:** An Enhanced Framework for Product Ranking using Deep Sentiment Analysis and Graph-Based Feature Modeling

**Objectives**

* To develop a robust data-driven framework that integrates advanced sentiment analysis and multicriteria decision-making to rank products based on online customer reviews, addressing interdependencies among product features and overcoming limitations such as implicit feature extraction and sentiment intensity.
* To employ cutting-edge natural language processing techniques, such as BERT and Graph Neural Networks, to enhance feature extraction by identifying both explicit and implicit product attributes from unstructured review data.
* To utilize a hybrid methodology combining deep sentiment analysis and Bayesian networks to assign weights to features, incorporating sentiment intensity and probabilistic interdependencies for more accurate and reliable product ranking outcomes.

**Outcomes**

The project will result in an improved product ranking system that provides more comprehensive and nuanced decisions compared to the original paper's framework. Key outcomes include: (1) increased feature coverage by 20-30% through implicit extraction, leading to better representation of customer concerns; (2) enhanced decision reliability by 10-15% via intensity-adjusted sentiment analysis, reducing biases from count-based approaches; (3) robust rankings validated for stability (e.g., Spearman correlation >0.9 in sensitivity tests); and (4) actionable insights for consumers and managers, such as ranked product lists with explainable reports on critical features (e.g., battery life impacting overall preference). The system will demonstrate superior performance on datasets like Amazon mobile reviews, enabling real-world applications in e-commerce recommendation systems.

**Modules (What Will Be Done in the Project)**

The project is structured around a five-phase modular architecture, implemented in Python with libraries like NLTK, Transformers (for BERT), mlxtend (for ARM), NetworkX (for FCM/BN alternatives), VADER, and NumPy/Pandas:

1. **Phase 1: Data Collection & Enhanced Feature Extraction** Load and preprocess OCR datasets (e.g., Kaggle's Amazon Product Reviews CSV). Extract explicit features using HAC (noun-adjective pairs) and implicit features using BERT (contextual embeddings for inferred aspects like "heat management" from "overheats"). Merge sets via semantic similarity (cosine threshold >0.8) with a loop for overlap <80% to ensure completeness.
2. **Phase 2: Feature Weighting with Interdependencies** Apply ARM (Apriori for rules) or alternatives like FP-Growth/DSA to mine patterns from merged features. Use FCM or Bayesian Networks to model causal graphs and compute dynamic weights, reflecting mutual influences (e.g., battery on price).
3. **Phase 3: Intensity-Aware Sentiment Analysis & Decision Matrix** Analyze sentiments with enhanced VADER (compound scores for intensity). Construct IVIF matrix incorporating membership/non-membership/hesitancy, scaled by intensity variance. Loop if variance >0.5 for data refinement.
4. **Phase 4: Product Ranking** Rank products using IVIF-MULTIMOORA (aggregating Ratio System, Reference Point, Full Multiplicative Form) on the weighted matrix.
5. **Phase 5: Sensitivity Analysis & Validation** Perturb weights to assess stability; validate accuracy (e.g., precision/recall >90%) with holdout sets. Generate reports and loop back if needed. Final output: Ranked products with visualizations (e.g., feature weight bars).

The project will be developed in a Python environment (Ubuntu/Windows, with virtualenv), using datasets like Kaggle's Amazon reviews instead of scraping, and tested on mobile phone data to mirror the paper's case study.