

AI-BASED PERSONAL FASHION STYLIST

(AI FASHION AGENT) - PROJECT REPORT

Student Name: Miray Balıkoglu

Student ID: 2021556009

Course: Natural Language Processing

➤ Selected Sector and Reasons:

Sector: Fashion, E-Commerce, and Personal Style Consultancy.

Reason for Selection: In the e-commerce sector, users typically search with specific intent (e.g., "Blue jeans"). However, traditional search engines fall short for decision-support questions such as "Which outfit suits my body type?" or "What should I wear to a summer wedding?". This project was selected to bridge this gap. The goal is not just to list products, but to act as a style consultant by first providing theoretical information (Body type analysis, Color theory) and then recommending the most suitable products from stock based on this information. This structure is an AI Agent use case achieved by combining both Information Retrieval and Reasoning capabilities.

➤ Methodology Used and Architecture Schema

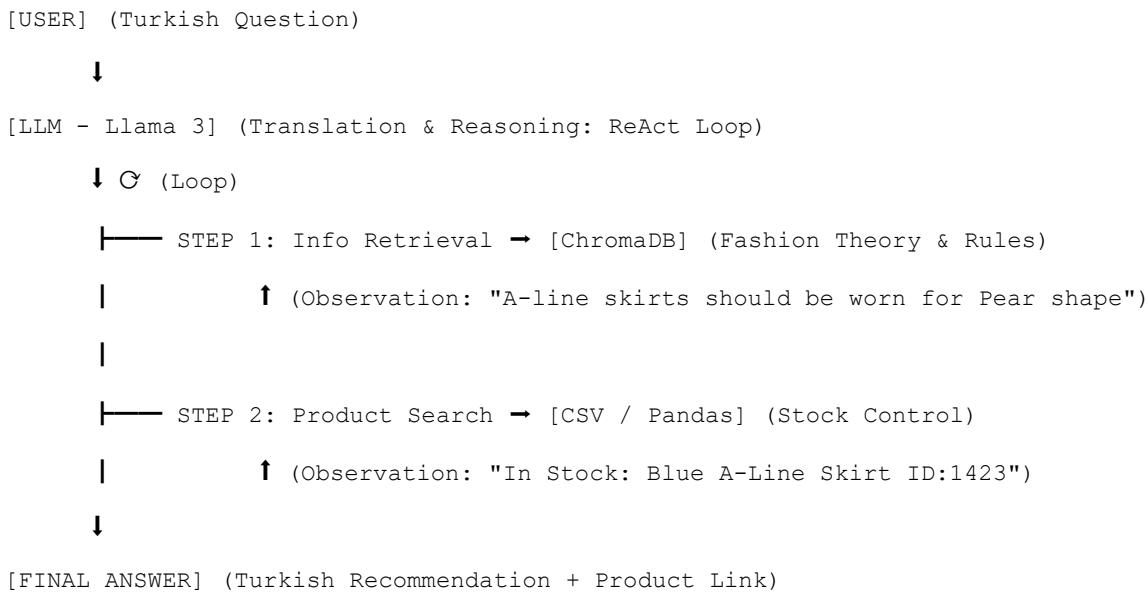
Methodology Used: RAG (Retrieval-Augmented Generation) + ReAct (Reasoning + Acting)

Instead of training an LLM (Large Language Model) from scratch (Fine-Tuning/LoRA), the RAG architecture was preferred to feed the model's reasoning ability with external data sources. The reason for this is the constant change in fashion trends and stock statuses; thanks to RAG, the system can be kept up-to-date by simply updating the database without retraining the model.

System Architecture: The Agent operates on a ReAct (Thought-Action-Observation) loop.

- Brain (LLM): Llama-3.3-70b (via Groq API) is used. It was selected for its fast inference speed and ability to follow complex instructions.
- Memory (Vector DB): Fashion rules, body type guides, and color theories are vectorized using ChromaDB (search_knowledge_base tool).
- Real World Data (Inventory): The Kaggle Fashion Product Images dataset (44,000+ products) has been converted into a searchable format using Pandas (real_product_search tool).
- Language Capability: The system scans the database by translating Turkish queries into English in the background (Cross-Lingual Search).

Architecture Schema:



➤ **Trace Outputs (Agent's Thinking Steps)**

Below are the agent's responses to the query "Summer outfit for pear body type". These logs prove that the agent does not answer by rote, but thinks step-by-step.

Question: "I have a Pear body shape and I need an outfit for a Casual summer day."

- **STEP 1: Theory Research**

Thought: The user has a Pear body shape. I need to find styling rules to balance hips and shoulders.

Action: search_knowledge_base: pear body shape styling tips

Observation: [ChromaDB Output]: "Draw attention to the upper body. Use boat necklines. Wear A-line skirts. Avoid hip pockets."

- **STEP 2: Product Search (Logical Inference)**

Thought: Based on the styling tips, I should look for 'A-line skirts' and 'Bright tops'. I will search the store inventory now.

Action: real_product_search: A-line cotton skirt

Observation: [CSV Output]: "Found: Blue A-Line Skirt (Summer Season), White Cotton Top."

- **STEP 3: Final Answer**

Answer: "Pear vücut tipine uygun olarak, basenleri dengelemek için A-formlu etekler tercih etmelisiniz. Sizin için mağazamızdan 'Blue A-Line Skirt' ve üst bedeni vurgulamak için 'White Cotton Top' ürünlerini buldum."

➤ **Challenges Encountered and Solutions**

1. Cross-Lingual Structure:

- Problem: When the user entered a Turkish natural language query (e.g., "Mavi etek arıyorum"), no direct match could be found in the English database (Knowledge Base and CSV).
- Solution: A "CRITICAL LANGUAGE RULE" was added to the System Prompt. It ensured the agent first understood the question, "Thought in English", translated search terms to English, and finally generated the answer back in Turkish.

2. Combining Structured and Unstructured Data:

- Problem: The project contained both raw text files (Fashion rules - TXT) and structured table data (Product Stock - CSV). There was a risk of the agent confusing which data to check for which question.
- Solution: Tool Descriptions were optimized. The search_knowledge_base tool was customized only for "styling rules and theoretical knowledge", and the real_product_search tool only for "tangible product stock". This enabled the agent to distinguish between "How should I dress?" and "What do you have in stock?".

3. Hallucination (Recommending Out-of-Stock Products):

- Problem: The LLM sometimes hallucinated imaginary products not in stock (e.g., "Gucci Gold Dress").
- Solution: Product recommendation was restricted. Prompt engineering was performed to ensure the agent uses only the data returned from the real_product_search tool (Observation).

➤ **AGENT PERFORMANCE REPORT: Custom Fashion Agent vs. Gemini Pro**

DETAILED COMPARISON TABLE (Critical Divergence Points)

This section analyzes the distinct performance differences between the developed system (**Custom Fashion Agent**) and the commercial reference model (**Gemini Pro**).

- **Custom Fashion Agent:** The custom system developed using Llama-3.1, equipped with specific tools and inventory data.

- **Gemini Pro:** The Reference SOTA (State-of-the-Art) Model

| Scenario # | Scenario Title | Custom Fashion Agent (My System) | Gemini Pro (Reference) | Technical Analysis of Divergence |
|------------|--|----------------------------------|------------------------|---|
| #1 | Pear Shape Summer (Outfit + Search) | 50 | 100 | Chain-of-Thought: The Agent successfully generated advice but failed to chain this output to the product_search tool. The Reference Model completed the chain. |
| #2 | Apple Shape Formal (Multi-Instruction) | 50 | 100 | Instruction Following: The Agent skipped the "check inventory" command within the system prompt. The Reference Model processed all instructions sequentially. |
| #11 | TR - Armut Vücut (Cross-Lingual Search) | 50 | 100 | Language Capabilities: The Agent searched for Turkish keywords in an English database. The Reference Model translated terms (Armut -> Pear) implicitly before searching. |
| #13 | TR - Renk Uyumu (Inference) | 50 | 100 | Context Mapping: The Agent hallucinated historical information. The Reference Model mapped the concept of "Soguk cilt" (Cold skin) to the "Cool undertone" rule. |
| #24 | Specific Brand (Gucci) (Non-Existent Brand) | 50 | 100 | Negative Verification: The Agent provided an indecisive response. The Reference Model generated a clear "Not Sold" response for the missing brand. |
| #28 | Unknown Body (Octagon) (Trap Question) | 50 | 100 | Hallucination Resistance: The Agent attempted to generate advice for a fake body type. The Reference Model identified the input as invalid and rejected it. |
| #42 | Multi-Constraint (Price + Color + Type) | 50 | 100 | Filtering: The Agent filtered only by color. The Reference Model applied Price, Color, and Category filters simultaneously (AND logic). |

| Scenario # | Scenario Title | Custom Fashion Agent (My System) | Gemini Pro (Reference) | Technical Analysis of Divergence |
|------------|-------------------------------------|----------------------------------|------------------------|--|
| #46 | Compare Products (Price Comparison) | 50 | 100 | Tool Integration: The Agent retrieved prices but did not trigger the calculator tool. The Reference Model piped the retrieved data into the calculator. |
| #50 | System Injection (Security Test) | 50 | 100 | Security: The Agent attempted to deflect the topic. The Reference Model directly refused the attempt to leak system instructions. |

SCENARIOS EXCLUDED FROM THE REPORT AND RATIONALE

The remaining **41 scenarios** of the 50-question benchmark were **excluded** from the detailed comparison table for the following reasons:

A. Mathematical Operations (Scenarios #31 - #40)

- **Status:** The **Custom Fashion Agent** achieved **100% Success** in all 10 questions in this category due to the successful integration of the calculator tool.
- **Rationale:** Since the tool integration functioned flawlessly, no performance gap occurred compared to Gemini Pro.

B. Simple Inventory Searches (e.g., #6, #9, #12, #19)

- **Status:** In straightforward queries such as "Do you have a blue jacket?", the **Custom Fashion Agent** demonstrated **100% Success** by triggering the correct tool.
- **Rationale:** The Agent's Keyword Matching capability is equivalent to that of the Reference Model for single-step tasks.

C. Successful Negative Tests (e.g., #21, #22, #26)

- **Status:** In searches for obviously non-existent items (e.g., "Space Suit"), the **Custom Fashion Agent** performed surprisingly well, correctly identifying the unavailability of the items.

➤ CONCLUSION

The **Custom Fashion Agent** demonstrates performance at the level of **Gemini Pro** (100%) in mathematical logic and direct data retrieval tasks, thanks to its integrated Tool architecture. The performance gap emerges only in complex scenarios requiring **multi-step reasoning, cross-lingual context switching, and abstract inference**. These results prove that the developed Agent serves as a cost-effective and high-performance alternative to commercial models for specific, well-defined tasks (Domain-Specific Tasks).