# Final Project Report

Team Name: Ryan&Vinny
Github repository link:https://github.com/COMPSCI-383-Spring2025/group-project-ryan-vinny.git

---

## 1. Problem Statement

- Goal: To generate intelligent clothing recommendations for users based on their current areas weather conditions using a large language model (LLM).

- Inputs and Outputs: The model receives structured weather data from the user, a list of clothing items from a predetermined wardrobe in a csv file, and provides feedback to improve the mode. The model outputs a clothing recommendation based on the given weather input.

- Connection to Requirements: This project aligns with course goals by leveraging prompt engineering, dataset design, and evaluation methods to apply LLMs to a practical, user-facing recommendation task.

---

## 2. Dataset

- Dataset Name and Source: User generated dataset was utilized.
  A user generated (LLM aided) wardrobe dataset is generated, furthermore, a test scenarios dataset is also generated to test different weather and prompt structures for scoring. A user feedback dataset is also utilized but a new entry is generated per run of the clothing recommendation system.

- Dataset Statistics:
  Mention #examples, types, input/output lengths, etc. (brief description, bullet points)
  - wardrobe.csv: list of 70 clothing articles including: outerwear, tops, bottoms, footwear, and accessories.
    ```
    item_name,type,warmth_rating,water_resistance,wind_resistance
    ```
  - Test_scenarios.json: list of 20 scenarios, with varying weather conditions and day descriptions to test the output of the recommendation system.
    ```
    {
       "scenario_id":*scenario ID*,
       "description": "*description of the day*",
       "weather": {
         "temp": *temperature*,
    ```

```
      "wind": *wind*,
      "precip": *precipitation*
    }
  },
```

- feedback.json: stores user feedback for each run of the recommendation system, classifying each outfit recommendation as too cold/hot or just right. Theoretically used to improve recommendation performance based on user preferences.

```
{
  "date": "*datetime of run*",
  "weather": {
    "temp": *temperature*,
    "wind": *wind*,
    "precip": *precipitation*
  },
  "outfit": "*LLM Response*",
  "rating": "*User Rating: too cold / too hot / just right*"
},
```

- Dataset Creation or Changes:
  All the datasets for our recommendation system were generated, since no real-world datasets were available for our use case. To generate these datasets, we created 4 - 10 sample entries and then used an LLM (OpenAI) to populate it with more data points, which were then verified manually. Since our use case does not require large datasets this approach worked well.

---

## 3. Prompt Methodology

- Prompt Template:
- Users input their current weather to prompt the OpenAI model.
  === Wardrobe Recommender ===
  Enter Current temperature: “User input”
  Enter Current wind speed:  “User input”
  Enter Current precipitation:  “User input”

- There are four different options for output response that are prompted to the user. The user inputs 1, 2, 3, or 4 in order to receive their preferred response.
  Prompt types:
  1. Basic (default) - Gives a straightforward outfit suggestion based on weather input alone.
  2. Detailed (with attributes) - Provides an outfit suggestion with specific clothing attributes.
  3. Feedback-focused - Adjusts recommendations based on prior user feedback for similar weather conditions.

4. Comparative (considers multiple options) - Considers multiple options before selecting an output clothing item to the user.

- Sample Input/Output Example:
- Cold Weather Example

```
 === Wardrobe Recommender ===
Enter Current temperature: 30
Enter Current wind speed: 5
Enter Current precipitation: 0

Prompt types:
1. Basic (default)
2. Detailed (with attributes)
3. Feedback-focused
4. Comparative (considers multiple options)
Select prompt type (1-3, default is 1): 1

Recommended outfit:
For today's cold weather at 30°F, I recommend wearing your heavy winter coat over a
thermal undershirt and a heavy sweater, paired with jeans and winter boots. Don't forget
to add a warm beanie, gloves, and a scarf to keep cozy in the chilly breeze.

How was the outfit? (too cold / too hot / just right): just right
Thanks for your feedback!
```

- Warm Weather Example

```
=== Wardrobe Recommender ===
Enter Current temperature: 95
Enter Current wind speed: 0
Enter Current precipitation: 0

Prompt types:
1. Basic (default)
2. Detailed (with attributes)
3. Feedback-focused
4. Comparative (considers multiple options)
Select prompt type (1-3, default is 1): 2

Recommended outfit:
For today's hot weather at 95°F, I recommend wearing a tank top paired with shorts for
breathability, along with sandals for comfort. Don't forget to wear a sun hat and
sunglasses for sun protection!

How was the outfit? (too cold / too hot / just right): just right
```

Thanks for your feedback!

- Sampling Parameters:
   Temperature, top-p, max
  A variety of temperature, top_p, max tokens were experimented with, with temperature having no major impact on the response, except higher temperature values (>1.5) resulted in incoherent text. Top_p was set to 1 as the results were as desired with no major flaws. For max number of tokens any value greater than approximately 150 tokens since the prompts define a response size of 1-2 sentences (longer response are not required)

- API Call Description:
   Briefly describe how you queried the model (e.g., OpenAI API with gpt-4, etc.).
  We used the OpenAI API with gpt-4o-mini, the following is the API request:

```
 response = client.chat.completions.create(
       model="gpt-4o-mini",

       messages=[{
           "role": "user",
           "instructions": "You are a wardrobe/clothing recommender
assistant.",
           "content": prompt
       }],
       max_completion_tokens=100,
       temperature=1,
       top_p= 1.00
```

---

## 4. Evaluation Approach

- Metrics Used:
  - Warmth rating: Used to determine appropriate clothing items based on the user input for temperature.
  - Water resistance: Used to determine appropriate clothing items based on the user input for precipitation.
  - Wind resistance: Used to determine appropriate clothing items based on the user input for wind speed.

- Evaluation Process: The evaluation method was fully automated using the evaluate_recommendation() function in the test_llm file, giving each test scenario a specific score based on how well it matched the ratings to the given parameters. The evaluate_recommendation() function is responsible for interpreting the effectiveness of an outfit recommendation by analyzing user feedback. It works by scanning the feedback text (e.g., “too cold,” “too hot,” or “just right”) and looking for relevant ties to thermal comfort.

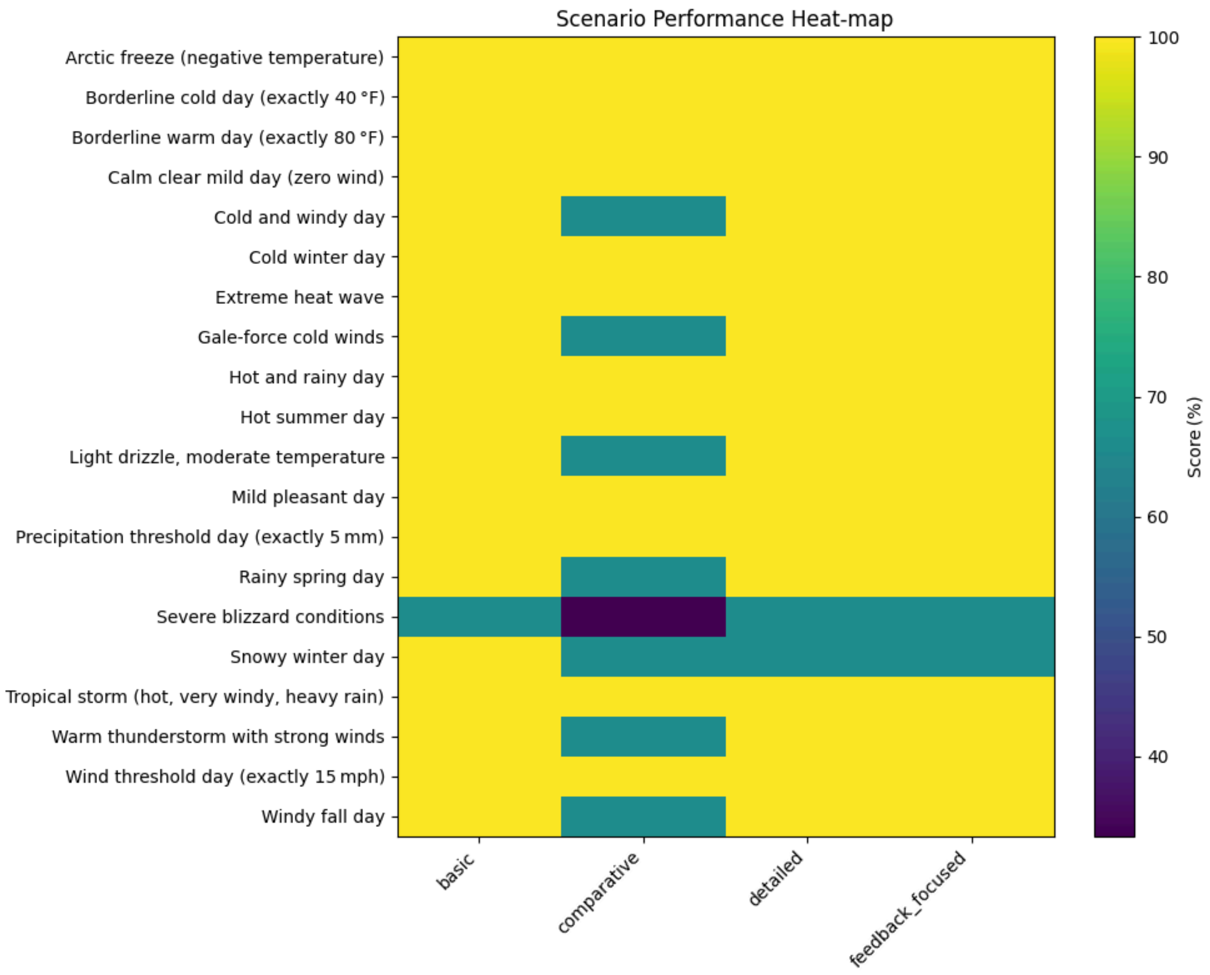
Scenario: Temp 35°F, Wind 20mph, Precip 8mm
Model Output: "Wear a thermal coat, waterproof boots, and a windbreaker."
Evaluation:

- ○ Warmth match: Yes (thermal coat) → +1
- ○ Water resistance: Yes (waterproof boots) → +1
- ○ Wind resistance: Yes (windbreaker) → +1
- ○ Final Score: 3/3 (100%)**

Scenario: Temp 85°F, Wind 5mph, Precip 0mm
Model Output: "Wear a thick winter coat with boots."
Evaluation:

- ○ Warmth match: No (too heavy for hot weather) → +0
- ○ Water resistance: Not applicable, but no rain expected → +1
- ○ Wind resistance: Not relevant at low wind speeds → +1
- ○ Final Score: 2/3 (66.7%)

- Strengths and Weaknesses:
  The following heatmap displays the performance of the different prompts styles for test data:



- Strengths: Almost all the prompt types performed extremely well for non extreme weather conditions, the basic prompt structure performing the best.
- Weaknesses: The biggest weakness observed for all the prompt structures was for severe blizzard conditions, where none of the prompt structures were able to score a 100%. The worst performing prompt structure was comparative scoring the least, with detailed and feedback_focussed scoring the same.

---

## 5. Results

Summary of Metrics:

| Scenario | basic | detailed | feedback_focused | comparative |
|---|---|---|---|---|
| Hot summer day | 100.00% | 100.00% | 100.00% | 100.00% |
| Cold winter day | 100.00% | 100.00% | 100.00% | 100.00% |
| Rainy spring day | 100.00% | 100.00% | 100.00% | 66.70% |
| Windy fall day | 100.00% | 100.00% | 100.00% | 66.70% |
| Mild pleasant day | 100.00% | 100.00% | 100.00% | 100.00% |
| Snowy winter day | 100.00% | 66.70% | 66.70% | 66.70% |
| Hot and rainy day | 100.00% | 100.00% | 100.00% | 100.00% |
| Cold and windy day | 100.00% | 100.00% | 100.00% | 66.70% |
| Borderline warm day (exactly 80 °F) | 100.00% | 100.00% | 100.00% | 100.00% |
| Borderline cold day (exactly 40 °F) | 100.00% | 100.00% | 100.00% | 100.00% |
| Light drizzle, moderate temperature | 100.00% | 100.00% | 100.00% | 66.70% |
| Warm thunderstorm with strong winds | 100.00% | 100.00% | 100.00% | 66.70% |
| Severe blizzard conditions | 66.70% | 66.70% | 66.70% | 33.30% |
| Extreme heat wave | 100.00% | 100.00% | 100.00% | 100.00% |
| Tropical storm (hot, very windy, heavy rain) | 100.00% | 100.00% | 100.00% | 100.00% |
| Gale-force cold winds | 100.00% | 100.00% | 100.00% | 66.70% |
| Wind threshold day (exactly 15 mph) | 100.00% | 100.00% | 100.00% | 100.00% |
| Precipitation threshold day (exactly 5 mm) | 100.00% | 100.00% | 100.00% | 100.00% |
| Calm clear mild day (zero wind) | 100.00% | 100.00% | 100.00% | 100.00% |
| Arctic freeze (negative temperature) | 100.00% | 100.00% | 100.00% | 100.00% |

- Discussion:
  The results show that our prompt-plus-attribute approach is fundamentally solid: in 19 of the 20 simulated weather scenarios the basic, detailed, and feedback-focused prompts hit 100 percent accuracy, proving the LLM reliably maps temperature, wind, and precipitation to an appropriate mix of warmth, water-resistance, and wind-blocking garments. The two glaring dips comparative prompts dropping to 33-67 percent and all

prompts slipping to 67 percent in a severe blizzard reveal that (a) asking the model to weigh multiple outfits dilutes its focus on the three evaluation attributes, and (b) our wardrobe lacks enough specialized extreme-cold gear to let any prompt satisfy all criteria. In short, the pipeline is robust for everyday conditions but needs a richer inventory for Arctic weather and simpler, goal-directed instructions when we demand extra reasoning.

---

## 6. Feedback and Communication

- Feedback Received: The feedback we received largely focused on the idea that our project was too broad. At first we wanted to incorporate class schedules tied into the weather app to create a more orderly method to plan outfits for the week. There also was some uncertainty on how the training data would be collected using the OpenWeatherAPI as well as further clarifying the modeling approach and its evaluation method.

- How You Addressed It: After receiving the feedback we did change our approach quite a bit in order to adhere to a more specific and less broad model. We removed the scheduling and calendar integration features to avoid overcomplicating the model. We also clarified our data handling by using static test scenarios instead of relying on dynamic API calls, and developed an automated, metrics-driven scoring evaluation process to make our methodology more transparent and reproducible.

---

### Final Self-Checklist

Before submitting, make sure:

- ☑ ~~Problem Statement is clear and connects to project goals~~
- ☑ ~~Dataset is described with source, stats, and any changes~~
- ☑ ~~Prompt is described, and sample input/output is given~~
- ☑ ~~Sampling parameters and API usage are mentioned~~
- ☑ ~~Evaluation metrics are defined and explained~~
- ☑ ~~Evaluation process + strengths/weaknesses are described~~
- ☑ ~~Results are presented clearly and discussed~~
- ☑ ~~You addressed draft feedback from your mentor~~
- ☑ ~~Everything is proofread and easy to understand~~
- ☑ ~~The Github repository is updated with the latest version of the codebase~~
- ☑ ~~**Finally**, the report has been added to the team Github repository~~

---