National College of Ireland

Configuration Manual

MSc Research Project
Msc in Artificial Intelligence

Sneha Mini Biju
Student ID: x23323701

School of Computing
National College of Ireland

Supervisor: Professor Abdul Shahid

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Sneha Mini Biju
Student ID:	x23323701
Programme:	MSc in Artificial Intelligence
Year:	2025
Module:	Practicum
Supervisor:	Professor Abdul Shahid
Submission Due Date:	September 1st, 2025
Project Title:	Configuration Manual
Word Count:	996
Page Count:	6

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Sneha Mini Biju
Date:	18th August 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Sneha Mini Biju
23323701

18th August 2025

1 System Requirements and Prerequisites

1.1 Hardware Requirements

Minimum System Specifications:

- **Processor**: Intel Core i5 or AMD Ryzen 5 (4+ cores)
- **RAM**: 8GB minimum, 16GB recommended
- **Storage**: 500MB available disk space (for data files)
- **Network**: Stable internet connection for data download

Recommended System Specifications

- **Processor**: Intel Core i7 or AMD Ryzen 7 (8+ cores)
- **RAM**: 16GB or higher (for large dataset processing)
- **Storage**: 2GB available disk space (SSD recommended for faster data access)
- **GPU**: Not required (CPU-based processing)

1.2 Software Requirements

Operating System:

- **Windows**: Windows 10/11 (64-bit)
- **macOS**: macOS 10.15 (Catalina) or higher
- **Linux**: Ubuntu 18.04 LTS or higher

Python Environment:

- **Python Version**: 3.8 or higher (3.9+ recommended)
- **Package Manager**: pip 20.0 or higher
- **Virtual Environment**: venv or conda

2 Installation and Setup

2.1 Environment Setup

Step 1: Python Installation

```
python --version
pip --version
```

Step 2: Create Virtual Environment

```
python --version
pip --version
```

Step 3: Create Virtual Environment(Distribution (2016))

```
conda create -n beauty_optimization_env python=3.9

# Activate virtual environment

conda activate beauty_optimization_env
```

Step 4: Install Dependencies

```
pip install -r requirements.txt

# Or install individually

pip install pandas==1.5.3
pip install numpy==1.24.3
pip install matplotlib==3.7.1
pip install seaborn==0.12.2
pip install TextBlob==0.17.1
pip install nltk==3.8.1
pip install scikit-learn==1.3.0
pip install reportlab==4.0.4
pip install PyYAML==6.0.1
pip install tqdm==4.65.0
```

2.2 Data Setup

Required Data Files:

- `All_Beauty.jsonl` – Raw Amazon beauty product reviews (311MB)
- Place in project root directory

Generated Data Files:

- `beauty_cleaned.csv` – Preprocessed review data (241MB)
- `sentiment_results.csv` – Sentiment analysis results (53MB)
- `rl_transitions_dataset.csv` – RL training dataset (781KB)
- `target_product_data.csv` – Target product information (9.1KB)

3.2 File Descriptions

preprocessing.py

- Reads raw JSONL data from Amazon
- Cleans and preprocesses review text
- Handles missing values and duplicates
- Converts timestamp to datetime format
- Computes word counts
- Exports cleaned CSV file

sentiment_analysis.py

- Performs TextBlob-based sentiment analysis
- Retrieves 7 beauty product aspects (scent, texture, packaging, effectiveness, price, longevity, application)
- Computes confidence weights from review attributes
- Calculates aspect-specific sentiment scores
- Exports sentiment results CSV

reinforcement_learning_improvement.py

- Runs Q-Learning algorithm for product improvement
- Defines 4D state space (5 levels in each dimension = 625 states)
- Implements 4 discrete actions (boost scent, texture, package, potency)
- Uses multi-component reward function with bonus and penalty
- Generates training reports and visualizations
- Supports both static and dynamic training modes

3 Configuration Parameters

3.1 Sentiment Analysis Configuration

Aspect Keywords (in sentiment_analysis.py):

```
beauty_aspects = {
    'scent': ['scent', 'smell', 'fragrance', 'aroma', 'perfume', 'odor'],
    'texture': ['texture', 'consistency', 'thick', 'thin', 'smooth', 'rough', 'creamy
    'packaging': ['packaging', 'bottle', 'container', 'tube', 'jar', 'design'],
    'effectiveness': ['work', 'effective', 'results', 'improve', 'help', 'fix'],
    'price': ['price', 'cost', 'expensive', 'cheap', 'value', 'worth'],
    'longevity': ['last', 'long', 'duration', 'stay', 'wear', 'persist'],
    'application': ['apply', 'easy', 'difficult', 'smooth', 'blend', 'spread']
}
```

Sentiment Discretization (in reinforcement_learning_improvement.py):

```
def discretize_sentiment(self, sentiment):
    if sentiment < -0.3:
        return 'low'
    elif sentiment < 0.0:
        return 'low-medium'
    elif sentiment < 0.3:
        return 'medium'
    elif sentiment < 0.6:
        return 'medium-high'
    else:
        return 'high'
```

3.2 Reinforcement Learning Configuration

State Space Configuration:

- **Dimensions**: 4 (scent, texture, packaging, effectiveness)
- **Levels per dimension**: 5 (low, low-medium, medium, medium-high, high)
- **Total states**: $5^4 = 625$ states
- **State representation**: [0, 1, 2, 3, 4] for each dimension

Action Space Configuration:

- **Actions**: 4 discrete actions
  - `improve_scent`
  - `improve_texture`
  - `improve_packaging`
  - `improve_effectiveness`

Training Parameters:

- **Episodes**: Configurable (default: 1000)
- **Learning rate (α)**: 0.02
- **Discount factor (γ)**: 0.95
- **Epsilon start**: 0.6
- **Epsilon end**: 0.1
- **Epsilon decay**: 0.04

Reward Function Components:

```
# Primary reward: improvement in targeted aspect
primary_reward = new_sent - old_sent

# Achievement bonus for significant improvements
achievement_bonus = 0.4

# Efficiency bonus for optimal actions
efficiency_bonus = 0.2

# Survival reward for maintaining performance
survival_reward = 0.1

# Penalty for worsening
penalty = -0.1

# Reward clipping to [-0.5, 0.5]
```

3.3 Preparing Data

Input format: JSONL format (Amazon review dataset)
Output format: CSV
Text cleaning: Convert all words to lowercase, remove duplicates, handle timestamps by converting them to datetime format.
Missing data: Eliminate rows with missing or incomplete values.

Sentiment Analysis Parameters:

- **Model**: TextBlob library
- **Confidence calculation**: Based on number of words, rating, and agreement on sentiment
- **Aspect extraction**: Keyword extraction with 7 beauty aspects
- **Output**: Weighted sentiment scores across products over time

4 Runtime Configuration

4.1 Command Line Execution

Data Preprocessing: `python preprocessing.py`
Sentiment Analysis: `python sentiment_analysis.py`
Data Exploration: `python data_exploration.py`
Poor Products Analysis: `python find_poor_products.py`
Reinforcement Learning Training (Last): `python reinforcement_learning_improvement.py`

5 Total Pipeline Runtime

5.1 Performance Optimization Impact

Optimization Type	Impact
Parallel Processing (4-8 cores)	
Sentiment Analysis	30–50% faster
Data Preprocessing	20–40% faster
Overall Pipeline	25–35% faster
Memory Optimization	
Chunked Processing	20–30% slower but more stable
Efficient Data Types	10–20% faster
Garbage Collection	5–15% slower but prevents crashes
Storage Optimization	
SSD vs HDD	2–5x faster I/O operations
File Compression	10–20% slower processing, 30–50% less storage

Table 1: Impact of various performance optimizations on the pipeline

5.2 Resource Usage Patterns

Category	Metric	Value / Pattern
4*CPU Usage	Preprocessing	60–80% (I/O bound)
	Sentiment Analysis	80–95% (CPU bound)
	RL Training	70–90% (CPU bound)
	Data Exploration	50–70% (mixed)
3*Memory Usage	Peak Memory	4–6 GB (during sentiment analysis)
	Average Memory	2–3 GB (throughout pipeline)
	Memory Growth	Gradual increase, peaks at sentiment analysis
3*Disk I/O	Read Operations	600 MB total (311 MB + 241 MB + 53 MB)
	Write Operations	300 MB total (241 MB + 53 MB + plots)
	I/O Pattern	Burst reads, gradual writes

Figure 1: Resource Usage Pattern

References

Distribution, A. S. (2016). Anaconda documentation. [Accessed 18 August 2025].
URL: *https://www.anaconda.com/docs/main*