

Configuration Manual: Personalized Dynamic Marketing Optimization: Integrating Hybrid Recommender Systems, Machine Learning and LLM concepts for E-commerce

MSc Research Project
Data Analytics

Amritha Riji Sudheer
Student ID: x23186887

School of Computing
National College of Ireland

Supervisor: Arjun Chikkankod

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Amritha Riji Sudheer.....
Student ID: x23186887.....
Programme: MSc. Data Analytics..... **Year:** 2024-25.....
Module: Research Practicum.....
Lecturer: Arjun Chikkankod.....
Submission Due Date: 15/09/2025.....
Project Title: Personalized Dynamic Marketing Optimization: Integrating Hybrid Recommender Systems, Machine Learning and LLM concepts for E-commerce.....
831.....
Word Count: **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: Amritha Riji Sudheer.....
Date: 10/08/2025.....

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):	

Personalized Dynamic Marketing Optimization: Integrating Hybrid Recommender Systems, Machine Learning and LLM concepts for E-commerce

Amritha Riji Sudheer
Student ID: x23186887

1 System Requirements

Component	Requirement
OS	Windows 10+, macOS10.15+ or Ubuntu 18.04+
Python Version	Python 3.8 – 3.11
GPU Support	Optional (NVIDIA CUDA-compatible GPU for NR-XGB-Mistral)
Model Support	Ollama Setup for using Mistral and Llama locally.
RAM	Minimum 8GB (16GB recommended for RL-LLaMA)
Disk Space	At least 5GB for datasets, models, and logs.

2 Python Dependencies

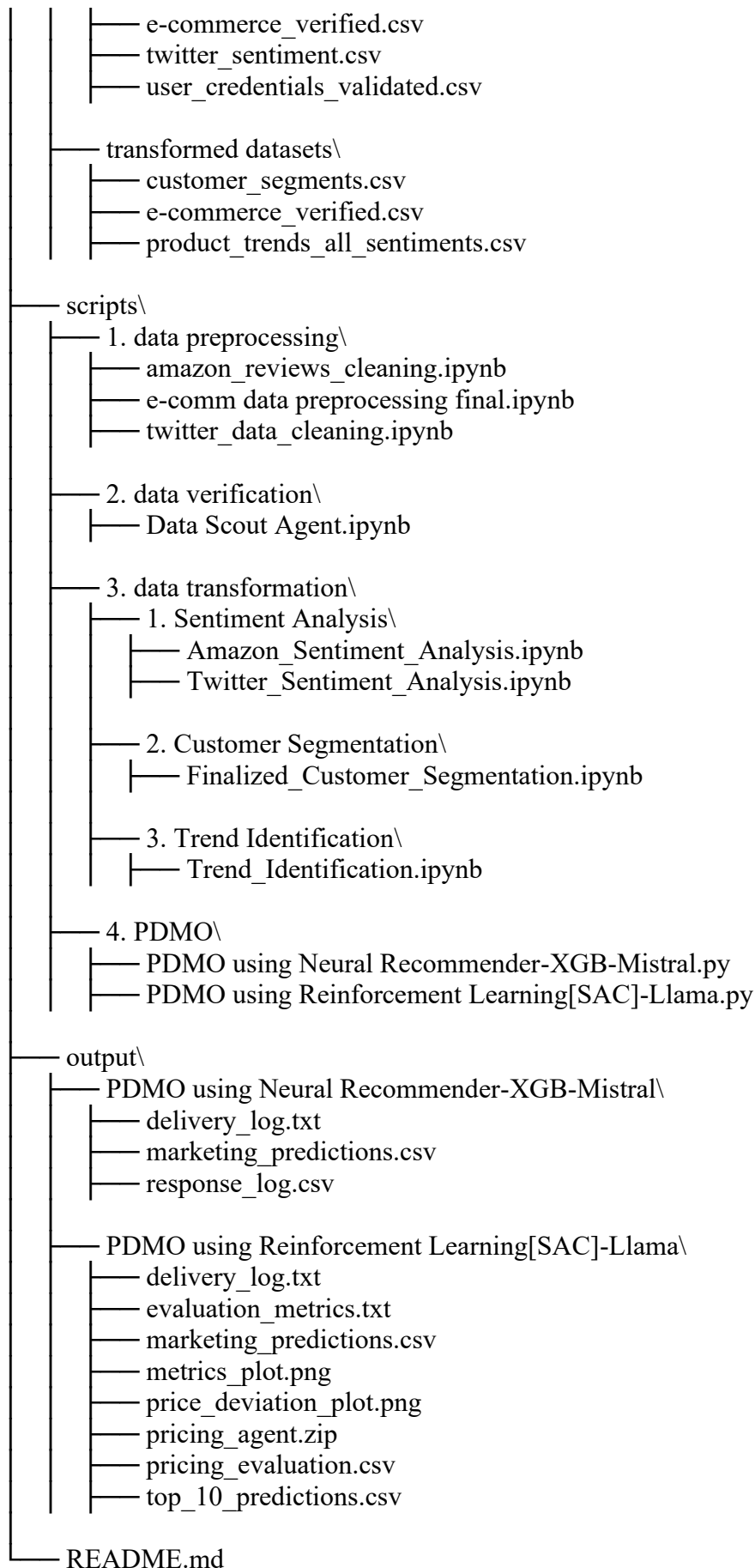
`pip install pandas numpy torch xgboost sentence-transformers transformers rapidfuzz scipy
scikit-learn nltk spacy psutil tqdm stable-baselines3 matplotlib seaborn ollama.`

`pip install ollama jupyter notebook
python -m spacy download en_core_web_sm`

Ollama framework required for LLaMA/Mistral models; install per Ollama documentation.
Hugging Face models (nlptown/bert-base-multilingual-uncased-sentiment, cardiffnlp/twitter-
roberta-base-sentiment) require internet access.

3 Folder Structure

```
PDMO CODES\  
├── datasets\  
│   ├── raw\  
│   │   ├── amazon_review.csv  
│   │   ├── ecommerce.csv  
│   │   └── twitter_sentiment  
│   └── processed\  
│       └── amazon_sentiment.csv
```



4 Model Input and Output Configuration

Input Details:

- `customer_segments.csv`: Contains CustomerID, Cluster (Low-Value, High-Value, Moderate High-Quantity, Premium), derived from K-Means clustering.
- `e-commerce_verified.csv`: Includes InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country, TotalPrice (Quantity $\times$ UnitPrice).
- `product_trends_all_sentiments.csv`: Provides product mentions with PositiveScore, NegativeScore, NeutralScore, NetScore (Positive - Negative).

Output Details:

- `delivery_log.txt`: Logs delivery status (e.g., "[2025-07-24 21:41:14] CustomerID: temp_109, Product: earphone, Price: \$10.04, Status: Delivered...").
- `marketing_predictions.csv`: Contains CustomerID, Cluster, Recommended_Product, Optimized_Price, Optimized_Message, Baseline/Optimized Conversion/Revenue.
- `response_log.csv`: Records CustomerID, Product, Opened, Clicked, Purchased, Revenue.
- **RL-LLaMA Specific:**
 - `evaluation_metrics.txt`: Summarizes metrics (e.g., MAE: 233.42, R²: -1.1318).
 - `metrics_plot.png`, `price_deviation_plot.png`: Visualizations of revenue, conversion, and pricing compliance.
 - `pricing_agent.zip`: Saved SAC policy model.
 - `pricing_evaluation.csv`: Detailed pricing metrics.
 - `top_10_predictions.csv`: Top 10 personalized recommendations.

Model Type	Input Files	Output Files
RL-LLaMA (SAC+LLaMA)	<code>customer_segments.csv</code> , <code>e-commerce_verified.csv</code> , <code>product_trends_all_sentiments.csv</code>	output/PDMO using Reinforcement Learning[SAC]- Llama/delivery_log.txt, marketing_predictions.csv, response_log.csv, evaluation_metrics.txt, metrics_plot.png, price_deviation_plot.png, pricing_agent.zip, pricing_evaluation.csv, top_10_predictions.csv
NR-XGB-Mistral (Neural Recommender + XGBoost + Mistral)	<code>customer_segments.csv</code> , <code>e-commerce_verified.csv</code> , <code>product_trends_all_sentiments.csv</code>	output/PDMO using Neural Recommender-XGB-Mistral/delivery_log.txt, marketing_predictions.csv, response_log.csv

5. Data Preprocessing Pipeline

1. **Amazon Electronics Reviews (10,000 rows, `amazon_sentiment.csv`):**

- **Source:** raw/amazon_review.csv.
 - **Sampling:** Stratified sample to preserve sentiment distribution.
 - **Cleaning:** spaCy (en_core_web_sm, parser/NER disabled) and NLTK pipeline: lowercase, remove URLs/emails/punctuation/special characters (regex), stopword removal, lemmatization.
 - **Output:** processed/amazon_sentiment.csv.
2. **SENTIMENT-140 Twitter Data (5,000 rows, twitter_sentiment.csv):**
- **Source:** raw/twitter_sentiment.
 - **Sampling:** Stratified sample, filtered for electronics (regex: 34 keywords, e.g., phone, laptop).
 - **Cleaning:** Custom function: lowercase, remove URLs/mentions/hashtags/punctuation (regex), tokenize (NLTK word_tokenize), lemmatize (WordNetLemmatizer).
 - **Output:** processed/twitter_sentiment.csv.
3. **E-Commerce Transactions (545 rows, 361 customers, e-commerce_verified.csv):**
- **Source:** raw/ecommerce.csv.
 - **Aggregation:** Summarize to 361 customers.
 - **Normalization:** Cast InvoiceDate to datetime (coerce errors), CustomerID to string (fill missing with "Unknown"), drop missing key columns, uppercase Description, remove whitespace.
 - **Filtering:** Case-insensitive regex (34 keywords) for electronics.
 - **Feature Engineering:** Calculate TotalPrice (Quantity × UnitPrice).
 - **Output:** processed/e-commerce_verified.csv, transformed datasets/e-commerce_verified.csv.
4. **User Credentials (user_credentials_validated.csv):**
- **Source:** processed/user_credentials_validated.csv.
 - **Purpose:** Validate customer data for segmentation.
5. **Segmentation:**
- Cap outliers in Quantity and TotalPrice (IQR method).
 - Normalize features with StandardScaler.
 - Apply K-Means (k=4, elbow method) on TotalPrice, Quantity, UnitPrice.
 - **Output:** transformed datasets/customer_segments.csv (Low-Value, High-Value, Moderate High-Quantity, Premium).
6. **Sentiment Analysis:**
- **Amazon Reviews:** nlptown/bert-base-multilingual-uncased-sentiment (Hugging Face), maps 1–5 star ratings to Positive (4–5), Neutral (3), Negative (1–2). Batch size: 16, truncate to 512 tokens.
 - **Twitter Data:** Use BERT, RoBERTa (cardiffnlp/twitter-roberta-base-sentiment), VADER; map outputs to Positive/Neutral/Negative. Batch processing with error handling.
 - **Output:** Appended to processed/amazon_sentiment.csv, processed/twitter_sentiment.csv.
7. **Trend Identification:**
- Filter Amazon/Twitter datasets by sentiment, extract product mentions (34 keywords).
 - Compute PositiveScore, NegativeScore, NeutralScore, NetScore (Positive - Negative).
 - **Output:** transformed datasets/product_trends_all_sentiments.csv with top 20 products.

6. Training Configuration

Parameter	RL-LLaMA (SAC)	NR-XGB-Mistral Recommender	(Neural NR-XGB-Mistral (XGBoost))
Epochs/Timesteps	3000 timesteps	50 epochs	N/A
Batch Size	N/A	64	N/A
Trees/Depth	N/A	N/A	100 trees, max depth 5
Optimizer	SAC (policy gradient)	Adam (lr=0.001)	Gradient Boosting
Loss Function	Reward-based	Binary Cross-Entropy	MSE
Learning Rate	N/A	0.001	0.1
Dropout	N/A	0.3	N/A

Message Generation: Mistral/LLaMA via Ollama, batch size 20, validated for word count (5–50), product/price inclusion.

7. Evaluation Metrics

- **Recommendation (NR-XGB-Mistral):** Accuracy (0.92), Precision (0.87), Recall (1.00), F1-Score (0.93), Cluster-Based Accuracy (0.59).
- **Pricing:** MAE (RL-LLaMA: 233.42, NR-XGB-Mistral: 3.55), MSE, R² Score, Price Deviation Compliance ($\pm 10\%$).
- **Message Quality:** Relevance (baseline/optimized: 0.99/0.97), Coherence (0.26–0.27), Perplexity (50.00), BLEU Score (0.35–0.54).
- **Campaign:** Open Rate (70.71–74%), CTR (38.57–44.59%), Conversion Rate ($\sim 1\%$), Total Revenue (\$42.10–\$324.98), Diversity Score (~ 0.75 –1.36).

8. Execution Instructions

Ensure input files (customer_segments.csv, e-commerce_verified.csv, product_trends_all_sentiments.csv) are in datasets/transformed datasets/.

Initialize Ollama using ollama serve in terminal.

Run PDMO scripts in scripts/4. PDMO/:

- RL-LLaMA: python "PDMO using Reinforcement Learning[SAC]-Llama.py"
- NR-XGB-Mistral: python "PDMO using Neural Recommender-XGB-Mistral.py"

9. Output Samples

Run PDMO scripts in scripts/4. PDMO/ to get the outputs:

- **RL-LLaMA:**
 - delivery_log.txt: Delivery status (e.g., "[2025-07-24 21:41:14] CustomerID: temp_109, Product: earphone, Price: \$10.04, Status: Delivered...").
 - marketing_predictions.csv: CustomerID, Cluster, Recommended_Product, Optimized_Price, Optimized_Message, etc.

- response_log.csv: CustomerID, Product, Opened, Clicked, Purchased, Revenue.
- evaluation_metrics.txt: Metrics summary (e.g., MAE: 233.42).
- metrics_plot.png, price_deviation_plot.png: Visualizations of revenue, conversion, pricing compliance.
- pricing_agent.zip: Saved SAC policy model.
- pricing_evaluation.csv: Detailed pricing metrics.
- top_10_predictions.csv: Top 10 personalized recommendations.
- **NR-XGB-Mistral:**
 - delivery_log.txt: Delivery status.
 - marketing_predictions.csv: CustomerID, Cluster, Recommended_Product, Optimized_Price, Optimized_Message, etc.
 - response_log.csv: CustomerID, Product, Opened, Clicked, Purchased, Revenue.

10. Known Limitations

- High Computational Resource Requirements: Use a system with at least 16GB RAM and a CUDA-compatible GPU. Monitor usage with psutil and enable garbage collection (gc.collect()) in scripts. For NR-XGB-Mistral, reduce batch size (e.g., from 64 to 32) in PDMO using Neural Recommender-XGB-Mistral.py. For RL-LLaMA, lower timesteps (e.g., to 1000) in PDMO using Reinforcement Learning[SAC]-Llama.py.
- Ollama and LLM Integration Challenges:
 - Start the Ollama server and verify model availability:
 - `ollama run mistral`
 - `ollama run llama3.1`
- Complex Dependency Setup

Install dependencies as specified:

- `pip install pandas numpy tqdm psutil rapidfuzz scipy scikit-learn xgboost torch sentence-transformers transformers nltk spacy matplotlib seaborn stable-baselines3 ollama jupyter notebook`
- `python -m spacy download en_core_web_sm`

Set up Ollama:

- `Ollama serve`
- `ollama pull mistral`
- `ollama pull llama3.1`