DeepMARCOSMobi: A Deep Learning-Enhanced Mobile Phone Ranking System

•*Name: Debamala Das Adhikari*
•*Registration No : 23122111*
5MScDsB

•*Guide: Dr. Lija Jacob*
•*Co-Guide: Miss. Margaret Salve*
Guide: Dr. Vandana Vijay Bhagat(Before)

Introduction

•Mobile phone ranking systems often use static methods, which may not solve to individual preferences effectively.
•This research focuses on developing a dynamic ranking system that adjusts feature weights based on user preferences using deep learning.
•By integrating the MARCOS method with deep learning, the system provides personalized rankings considering factors like gaming, camera, and budget.
•The goal is to offer a more accurate and user-centric mobile phone ranking system.

Goal of the Domain:

•**Improve Mobile Phone Ranking Accuracy:** Create a more reliable and personalized ranking system that adapts to user preferences.

•**Dynamic User Preferences Integration:** Use deep learning to adjust feature weights dynamically, considering factors like gaming performance, photography, and budget.

•**Enhanced Rankings via Deep Learning and MARCOS:** Incorporate the MARCOS method for multi-criteria decision making, combined with deep learning for better weight adjustment, offering accurate and relevant rankings.

Applications of the Domain

•**E-commerce:** Provide personalized mobile phone recommendations to users based on their preferences enhancing shopping experiences.

•**Retail Industry:** Assist businesses in identifying and stocking popular smartphones that align with consumer demands and preferences.

•**Tech Reviews:** Help reviewers generate rankings based on objective criteria and tailored recommendations, aiding users in making informed purchase decisions.

Literature Review:

- **MCDM Techniques:**
  Several Multi-Criteria Decision-Making (MCDM) methods like TOPSIS, AHP, and MARCOS have been used for mobile phone ranking. MARCOS offers a more flexible and accurate approach by considering ideal and anti-ideal solutions, which aligns well with user preferences.
- **Deep Learning for Ranking:**
  Deep learning models have shown promise in adjusting ranking weights dynamically based on user preferences and patterns, providing a personalized and context-aware ranking system. Various studies have explored neural networks to predict feature importance for ranking systems.
- **LLMs in Recommendation Systems:**
  Large Language Models (LLMs) like GPT, BERT, and others have been integrated into recommendation systems to generate personalized suggestions, refine rankings, and provide explanations based on natural language queries.
- **Research Gaps:** Despite progress, existing methods often lack the dynamic adjustment of ranking criteria based on real-time preferences, and few studies combine deep learning-based weight adjustment with LLM-based explanations in mobile phone ranking systems.

Problem Statement

Existing Ranking Systems:

- Rely on static weight assignments for criteria.
- Lack adaptability to individual user preferences (e.g., gaming, photography, budget).
- Do not dynamically adjust rankings based on evolving user needs.

Challenges:

- Inaccurate recommendations due to rigid, predefined feature weights.
- Users may not find rankings that align with their specific priorities.

Need for an Intelligent System:

- Dynamic Weight Adjustment: Incorporating deep learning models to adjust feature weights based on user preferences.
- Personalized Rankings: Tailoring phone rankings to individual preferences and needs.

Research Objectives

- **Develop a Dynamic Mobile Phone Ranking System**:
Leverage the MARCOS (Multi-Attribute Realizing the Comprehensive Optimization Solution) method for more accurate rankings.
- **Integrate Deep Learning for Weight Adjustment**:
Use deep learning models to predict and adjust feature weights based on individual user preferences (e.g., gaming, photography, budget).
- **Incorporate Large Language Models (LLMs) for Personalized Explanations**:
Provide transparent, customized recommendations and reasoning behind rankings.

Dataset & Attributes

Data Source:

Gigasheet

Dataset Description: Real World Smartphone Dataset

Overview,the dataset consists of 975 entries, each representing a smartphone with various features.

Category	Features
Basic	Brand, Model, Price, Average Rating
Performance	Processor, RAM, Internal Storage
Display	Screen Size, Refresh Rate, Resolution
Camera	Rear & Front Camera Specs
Battery	Capacity, Charging Speed
Connectivity	5G Support, Network Bands

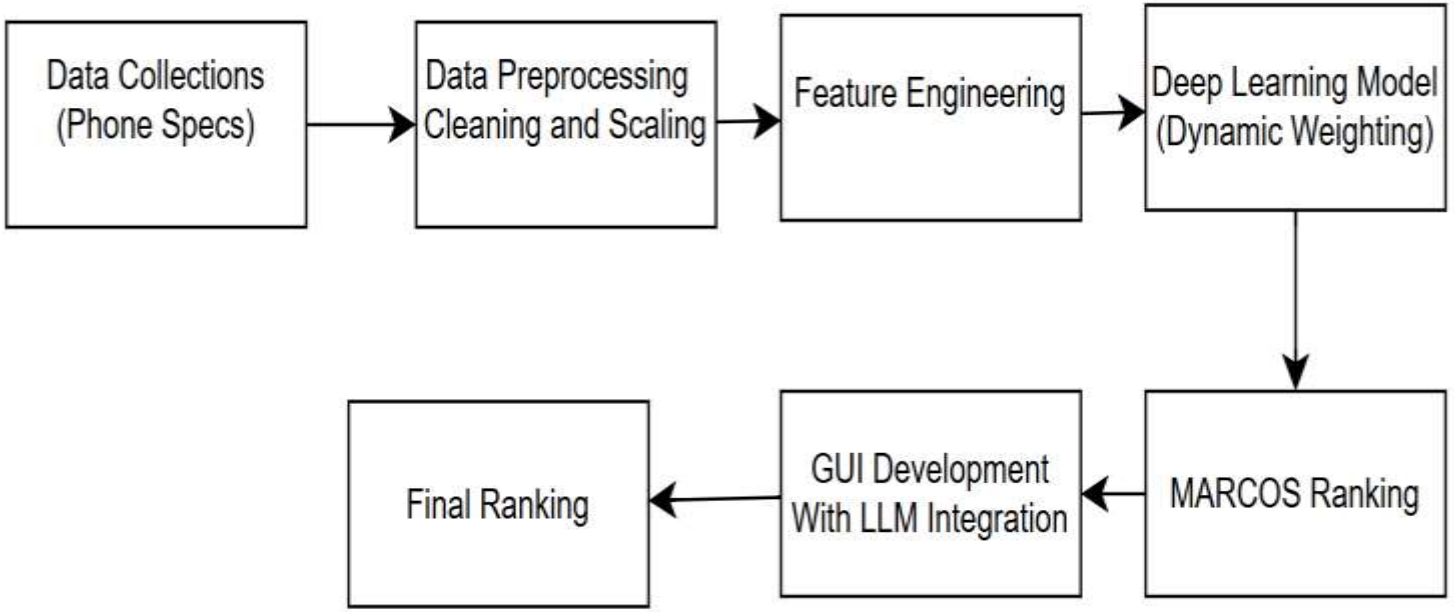
Attributes:

1. **brand_name** *(string)*: The brand of the smartphone (e.g., Apple, Samsung, Xiaomi).
2. **model** *(string)*: The specific model name or identifier.
3. **price** *(integer)*: The price of the smartphone (in an unspecified currency).
4. **avg_rating** *(float)*: The average user rating for the smartphone.
5. **5G_or_not** *(integer)*: Indicates if the phone supports 5G (1: Yes, 0: No).
6. **processor_brand** *(string)*: The brand of the processor (e.g., Qualcomm, MediaTek, Bionic).
7. **num_cores** *(integer)*: The number of cores in the smartphone's processor.
8. **processor_speed** *(float)*: The speed of the processor in GHz.
9. **battery_capacity** *(integer)*: The battery capacity in milliampere hours (mAh).
10. **fast_charging_available** *(integer)*: Indicates if fast charging is supported (1: Yes, 0: No).
11. **fast_charging** *(integer)*: The power of fast charging in watts.
12. **ram_capacity** *(integer)*: The RAM size in gigabytes (GB).
13. **internal_memory** *(integer)*: The internal storage size in gigabytes (GB).
14. **screen_size** *(float)*: The diagonal size of the screen in inches.
15. **refresh_rate** *(integer)*: The screen refresh rate in Hertz (Hz).
16. **num_rear_cameras** *(integer)*: The number of rear cameras on the phone.
17. **os** *(string)*: The operating system of the smartphone (e.g., Android, iOS).
18. **primary_camera_rear** *(integer)*: The megapixel value of the primary rear camera.
19. **primary_camera_front** *(integer)*: The megapixel value of the primary front camera.
20. **extended_memory_available** *(integer)*: Indicates if the phone supports expandable storage (1: Yes, 0: No).
21. **resolution_height** *(integer)*: The height of the screen resolution in pixels.
22. **resolution_width** *(integer)*: The width of the screen resolution in pixels.

Flow of the Research

- **Data Collection**
  Collect smartphone features.
- **Feature Engineering**
  Preprocess and normalize data for consistent comparison and scoring.
- **Model Development**
  Implement MARCOS for initial ranking and integrate deep learning models to predict dynamic feature weights based on user preferences.
- **Deployment & GUI Integration**
  Develop a user-friendly GUI using Tkinter for real-time ranking adjustments and explanations, integrating LLMs for personalized insights.

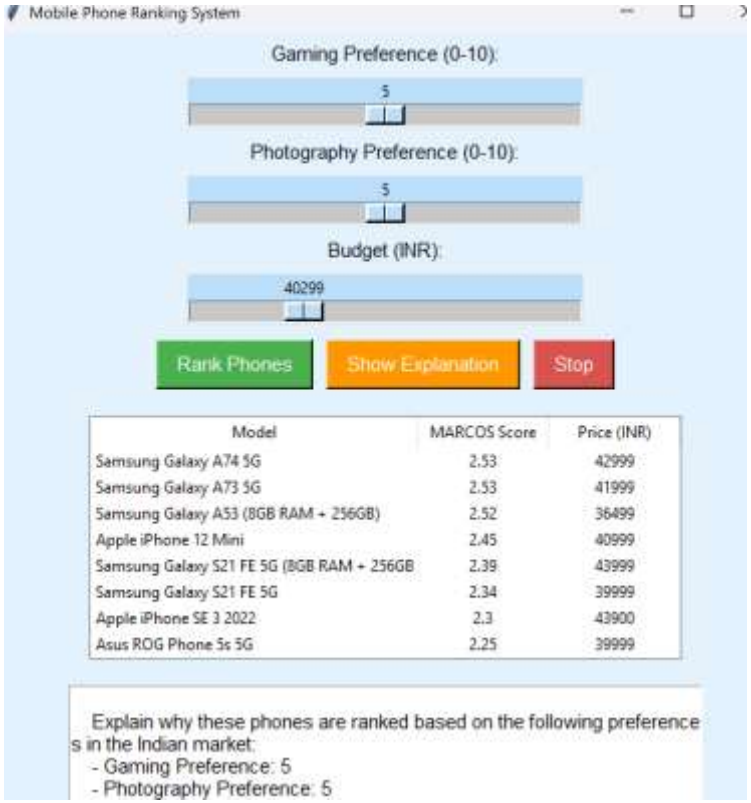
Implementation Flow

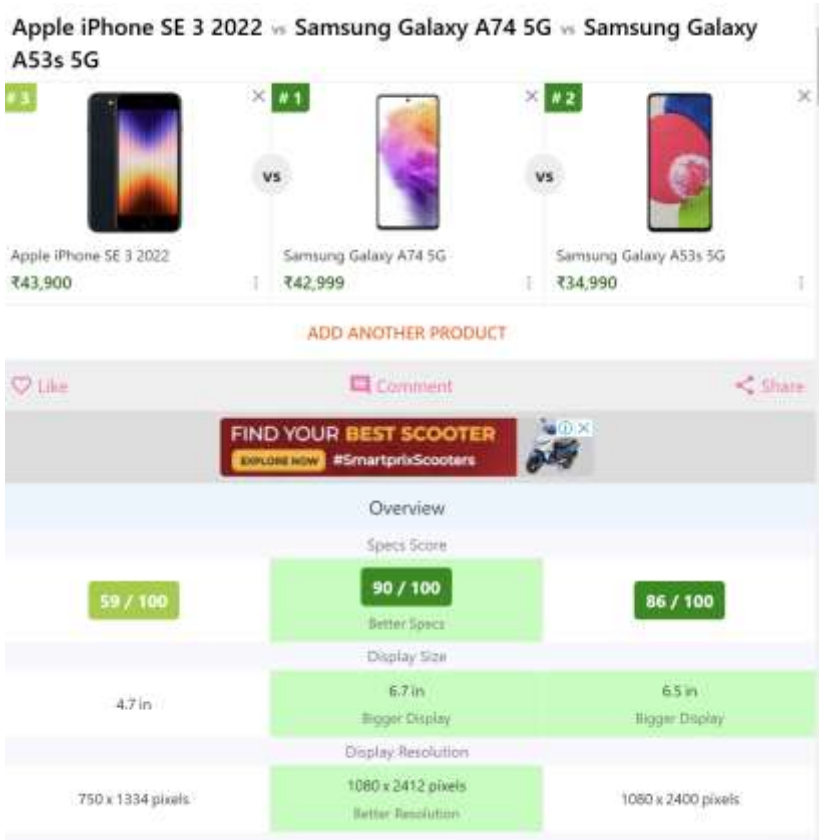


Methodology

Data Preparation:
Smartphone features are selected and scaled (e.g., gaming, camera, price).
Simulated dynamic weights (target variable) are generated for training.
Deep Learning Model:
A neural network is built with layers like Dense, BatchNormalization, and Dropout.
The model learns dynamic feature importance by predicting weights for each feature.
Model Training:
The model is compiled using Adam optimizer with mean squared error loss.
It is trained over 150 epochs to adjust weights based on smartphone features.
MARCOS Scoring:
Step 1: Adjust feature scores by multiplying with predicted weights.
Step 2: Calculate Euclidean distances to ideal best and worst solutions.
Step 3: Compute MARCOS score to rank smartphones based on proximity to ideal solutions.
Ranking:
Smartphones are ranked in descending order of MARCOS scores, with the highest score indicating the best option for the user.

Results And Comparison From A Site





Findings:

I selected three different phones based on the rankings generated by **DeepMARCOSMobi** (the model implemented by me) and compared them with the rankings from **Smartprix.com** to verify consistency.

1. **Samsung Galaxy A74** ranked **first** in both my model and Smartprix(an online website).
2. **Samsung Galaxy A53s 5G** ranked after the first one in my model and also held a similar position on Smartprix.
3. To test for potential bias toward Android, I included the **Apple iPhone SE3**, which ranked after this two in my model, aligning with its position on Smartprix.

Conclusion

The alignment between **DeepMARCOSMobi** rankings and **Smartprix.com** validates the model's accuracy and reliability. Additionally, since **Apple iPhone SE3** ranked appropriately without being unfairly penalized, it suggests that the model is **not biased toward Android** and fairly evaluates all brands based on their features.

Conclusion

•**Key Contributions**:
•Developed a novel mobile phone ranking system using a deep learning-based Multi-Criteria Decision-Making (MCDM) approach with MARCOS.
•Integrated dynamic weight adjustment based on user preferences to offer personalized recommendations.
•Leveraged LLMs (GPT-2) to generate insightful explanations of ranking decisions, enhancing transparency.
•**Impact**:
•The system helps users make more informed decisions by ranking mobile phones based on their specific needs and preferences (e.g., gaming, photography, budget).
•Improved ranking accuracy with the ability to dynamically adjust to changing user inputs and market trends.

Future Work

- **Real-Time Data Integration**:
  - Incorporate live reviews and social media sentiment for up-to-date recommendations.
  - Use real-time trend analysis for dynamic and context-aware recommendations.
- **LLM Fine-Tuning**:
  - Fine-tune large language models (LLMs) to better understand user preferences and provide context-specific recommendations.