Design and Implementation of a Cross-platform mobile-commerce Recommender System using TensorFlow Recommenders, Express.js, Flutter, and Docker Containerization

Osama Mohammed Ahmed[1], Anas Maan Tahir Maaroof [1]

[1]Department of Studies and Planning, Mosul University

Abstract: In mobile commerce, recommendation systems play a pivotal role in bolstering user engagement and fostering business expansion. This study outlines the development of a recommendation system, employing a combination of TensorFlow models, Docker containers, Node.js, and Flutter, tailored specifically for the domain of book recommendations.

The challenges inherent in constructing and deploying recommendation systems for both mobile and web applications are systematically addressed in this research. The architecture leverages TensorFlow Recommenders to formulate two essential models: retrieval and ranking. To seamlessly integrate these models into applications, a user-friendly RESTful API is established using Node.js. Additionally, the ongoing development of a prototype mobile application with Flutter demonstrates the practical implementation of this recommendation system.

A noteworthy achievement of this paper is the encapsulation of the RESTful API, along with TensorFlow models, within Docker containers as a single deployable image and package. This containerization simplifies the deployment process, enabling the API to be hosted and accessed effortlessly from various platforms, including mobile and web applications, through a designated endpoint.

This work amalgamates a diverse range of cutting-edge technologies to fashion a straightforward, yet powerful recommendation system tailored to the unique demands of mobile commerce within the book domain. It underscores the intricacies of recommendation systems and their pivotal role in enriching user experiences. By showcasing the effective utilization of TensorFlow Recommenders, Docker, Node.js, and Flutter, this research offers a practical and robust solution to the complex challenge of constructing, deploying, and seamlessly integrating recommendation systems into the ever-evolving landscape of mobile commerce.

Keywords: Recommender System, M-Commerce, TensorFlow, Node.js, Flutter

I. Introduction

In today's digital world, recommendation systems are essential for helping users discover relevant content or products. In the mobile commerce space, personalized shopping experiences are critical for enhancing user engagement and driving business growth.

A recommender system is a tool that analyzes user data to predict what other products they might be interested in. This is done by analyzing the user's past behavior, such as the items they have purchased or rated, and the items that other users with similar interests have purchased or rated (Anandaraj et al., 2021).

Recommendation systems can be implemented using a variety of methods. Some of the most common methods include collaborative filtering, content-based filtering, and knowledge-based filtering. Collaborative

DOI:http://doi.org/10.24086/cocos2024/paper.1473

filtering methods recommend items to users based on the ratings and preferences of other users who have similar interests, Content-based filtering methods recommend items to users based on the content of the items and the user's profile. Knowledge-based filtering methods recommend items to users based on a knowledge base of user preferences and item features (Fudholi et al., 2021).

Recommendation systems (RS) frequently appear in e-commerce environments, where websites suggest products and services that align with users' preferences, relying on user opinions as a basis (Venkatrama, 2017).

Mobile commerce, or m-commerce, is the buying and selling of goods and services through mobile devices. It is an extension of e-commerce that allows people to make purchases from anywhere, using their smartphones or tablets (Akanferi et al., 2022).

TensorFlow is a popular open-source software library for machine learning, especially deep learning. It is used by many technology giants, such as PayPal and eBay. TensorFlow was developed by Google Brain, and it is based on the dataflow programming paradigm. It allows users to define and train machine learning models using a variety of programming languages, including Python, C++, and Java. TensorFlow is a powerful tool that can be used for a variety of machine learning tasks, including image classification, natural language processing, and speech recognition. It is also used for research in machine learning and artificial intelligence (Filus & Domańska, 2023).

The problem: Designing an effective recommendation system and seamlessly embedding it into mobile and web applications involves a range of skills. This includes knowledge of machine learning, programming, math, and handling large datasets.

The objective: Our research aims to make things easier by suggesting a model that uses easy-to-use and flexible technologies to smoothly integrate the recommendation system into cross-platform apps.

To achieve this, we rely on the TensorFlow Recommender library, which is user-friendly and powerful, to create retrieval and ranking models. TensorFlow makes it easy to build and train models efficiently.

To make integration straightforward, RESTful API has been developed using Node.js. This API serves as a user-friendly connection point for the recommendation system and the Flutter framework, simplifying the integration process.

To ensure compatibility across different platforms, we use Docker containers to deploy these models (retrieval and ranking), This approach simplifies the deployment process and ensures the system works consistently.

Our Contribution: Our approach allows developers and business owners to create versatile recommendation system compatible with Android, iOS, and web platforms. It simplifies the complexities of recommendation system development, offering a practical solution for developers and researchers.

II. Related Works

(Anandaraj et al., 2021), introduce A book recommendation technique serves as a tool employed by e-commerce websites, with Incorporating TensorFlow techniques ensures the stability and efficiency of a recommendation system, allowing it to suggest the most relevant books to users in a straightforward and effective manner.

In the realm of fashion retail, a recommendation system based on multi-level clustering of items and user profiles across online and physical stores has been proposed. This solution was developed as part of the Feedback project, supported by Regione Toscana, and tested with real data from Tessilform, a retail company. Over a one-year period, from December 2019 to December 2020, this recommendation tool was found to stimulate customer interest, resulting in a notable 3.48% increase in purchases. This study demonstrates the practical benefits of recommendation systems in the fashion retail sector (Bellini et al., 2023).

In this paper (Tahir et al., 2021), a method is being presented for the construction of an apparel recommendation system. The modeling and construction phases of an e-commerce apparel recommendation system have been defined. The prediction will be based on content-based filtering, which is the most popular type of filtering system.

DOI:http://doi.org/10.24086/cocos2024/paper.1473

These systems rely on a user's ratings to construct a profile and obtain preliminary information about a user to avoid not knowing a new user.

(Fudholi et al., 2021), in this study, A mobile tourism recommendation system has been developed, offering destination suggestions based on past travel photos. To create image classifiers for tourism, utilized the state-of-the-art image classification framework, EfficientNet, with a focus on the EfficientNet_Lite variant. These models were trained in the tourism context using TensorFlow Lite. In our experiments, we observed high accuracy across different EfficientNet_Lite architectures. The top-performing model, EL-0, achieved 96% accuracy on our training dataset, 87% on the validation dataset, and 85% on the testing dataset.

addresses challenges in E-commerce development and enhances traditional algorithms by incorporating collaborative filtering. This approach, widely employed domestically, proves valuable, especially with limited user evaluation data (Lou, 2022)

(Xu & Wang, 2022) has Developed an intelligent recommendation system using social awareness and mobile computing. Analyzed the behavioral characteristics of current E-commerce product recommendation systems and constructed one using mobile computing data processing techniques. Test results demonstrate that this system outperforms traditional E-commerce product recommendation systems. It accelerates users' product discovery, enhances recommendation accuracy, and significantly reduces recommendation errors compared to traditional methods.

(Zhu, 2022) This article introduces a method for recommending mobile commerce services in tree-based networks by fusing data from multiple sources. To achieve efficient storage and access of structured data, we store tree-node relationships using redundant data, thereby completely decoupling tree-type nodes and reaching the tree structure. Given the limitations of multimedia mobile devices, such as limited hardware, storage, and computing power, we utilize client-side caching of partial data and on-demand sequential server data requests. This approach leverages the hierarchical tree structure to design an efficient system. The tree synchronization model enables mobile devices to efficiently buffer and access server data.
(Guo et al., 2018) Introduces a method leveraging multi-source information to analyze consumer preferences in E-commerce recommendation systems, particularly tailored for mobile E-commerce. This approach incorporates an enhanced radial basis function (RBF) network for recommendation weight determination and an improved Dempster-Shafer theory for multi-source information fusion. Experimental outcomes demonstrate that the proposed method outperforms traditional approaches in terms of recommendation accuracy, simplicity, coverage rate, and recall rate.

(Kamble et al., 2021) Designed a music recommendation system utilizing reinforcement learning principles. Employed the Deep Deterministic Policy Gradient (DDPG) model, implemented using TensorFlow libraries, in conjunction with Firebase and BigQuery Google Cloud Services. These services were utilized for storing and aggregating user preferences within the app, specifically, likes and dislikes. The DDPG model, functioning as a Markov Decision Process, leveraged Deep Reinforcement Learning to autonomously acquire the optimal policy for item recommendations.

In this study (Kantepe et al., 2020) The MovieLens dataset served as the foundation, and a recommendation system was crafted employing deep learning techniques, specifically autoencoders. Throughout the recommendation system's development, various optimization algorithms, including Gradient Descent, Stochastic Gradient Descent, RmsProp, and Adam, were explored on the TensorFlow platform within the Python programming language. This exploration aimed to enhance the system's overall performance while also investigating how the surge in data volume affected these optimization algorithms. Consequently, it was ascertained that the most effective optimization algorithm was Adam, resulting in a test score error of 1.363. Furthermore, it was noted that diminishing the gap ratio within the training set correlated with a reduction in the test score error.

This paper (Lu & Liu, 2023) has employed the classic DNN double tower recommendation algorithm within the recommendation system as its ranking algorithm. It creates separate embeddings for users and items and constructs the network model using TensorFlow. This system's design aligns with contemporary needs, enabling individuals to benefit from recommendation system applications in the commodity industry. It represents an efficient and innovative solution.

DOI:http://doi.org/10.24086/cocos2024/paper.1473

(Venkatrama, 2017) In this paper, a novel approach has been introduced in which business intelligence (BI) concepts are applied to recommendation systems (RS) to enhance their ability to adapt to user changes and complex business conditions. A BI-based framework called a Business Intelligent Recommender System (BIRS) is proposed, which leverages techniques like Online Analytical Processing (OLAP) and data mining to improve RS performance and offer more personalized recommendations. The framework's application in a B2C e-commerce scenario is also demonstrated.

Limitations/ the Gap: Upon an examination of the related literature, it becomes evident that there lacks a standardized and straightforward model for implementing recommendation systems that seamlessly integrate with cross-platform applications. While some studies have utilized TensorFlow tools to develop recommendation systems, none have harnessed the specialized TensorFlow Recommenders library explicitly designed for this purpose. Furthermore, the methodologies employed for deploying recommendation models to function with mobile or web applications are often complex, and the utilization of modern containerization systems like Docker is absent.

Consequently, the identified gaps in the current body of research are as follows:

1. The complexity involved in implementing recommendation systems without the use of specialized tools tailored for this task like TensorFlow Recommenders.
2. The absence of standardized methods for deploying recommendation models as integral components of mobile and web applications, including the underutilization of containerization technologies such as Docker and RESTful APIs.
3. Limited cross-platform compatibility, with recommendation systems typically designed for either web, mobile, or cloud environments, lacking versatile solutions like API integration with Flutter.

III. Methodology

Our recommendation system is designed to be easily integrated into cross-platform applications. Here is a high-level overview of the system's architecture:

1. Data Collection and Preprocessing

The dataset was taken from (Amazon Books Reviews), This study utilized a dataset from Kaggle (*Kaggle Datasets*, n.d.), focusing on user's reviews and ratings for books on Amazon. The dataset contains:

- User ID: A unique user identifier.
- Book title: the title of the books.
- Ratings: users satisfaction scores, ranging from 1 to 5.

Two CSV files were employed in the experiment, namely, "ratings.csv" and "books.csv." "ratings.csv" contains over one million records with columns for user ID, book title, and ratings. On the other hand, "books.csv" contains more than 200,000 unique book titles, The CSV files are securely stored in Google Drive, an additional cloud-based storage solution, eliminating the need for repetitive uploads during experiments while working with Google Colab.

During the data preparation phase, both CSV files underwent cleaning processes to remove spaces, blank cells, and special characters using Python.

It's worth noting that, due to limitations in computer resources, cloud resources, and potential internet connectivity issues, only a subset of the data was utilized for the experiment. Specifically, 100,000 user ratings were used for analysis and model training in Google Colab.

2. TensorFlow Recommender Library

TensorFlow Recommenders (TFRS) is a library that helps you build recommender systems end-to-end. It provides tools for data preparation, model formulation, training, evaluation, and deployment. TFRS is built on Keras, which makes it easy to use even if you are not familiar with machine learning. However, it also provides enough flexibility to build complex models, we created and trained two models, the Retrieval model and the Ranking model using Google Colab, an online platform for Python and TensorFlow.

DOI:http://doi.org/10.24086/cocos2024/paper.1473

Google Colab is a free, cloud-based platform that allows users to run Python code, especially for machine learning and data analysis. It provides access to GPUs and TPUs, which are powerful processors that can speed up the training of machine learning models. Colab also comes with pre-installed libraries, so users don't have to worry about installing them themselves. Additionally, Colab allows users to collaborate on projects, making it a great tool for teams. Overall, Google Colab is a powerful and versatile platform that is ideal for data science tasks.

In the Retrieval model: we utilized a two-towers model, which signifies a modern approach to training recommendation system models. This technique incorporates two crucial components: the query model (in our case, it corresponds to the user ID) and the candidate model (representing the product name or product ID). Following this, we employed a deep neural network to create embeddings for both the query and candidate models. These embeddings capture essential features and representations of the user and product data. Subsequently, we computed the dot product between these embeddings, which serves as a measure of similarity between the user and the product. This similarity score is a key factor in making relevant product recommendations to the user (see Figure 1).

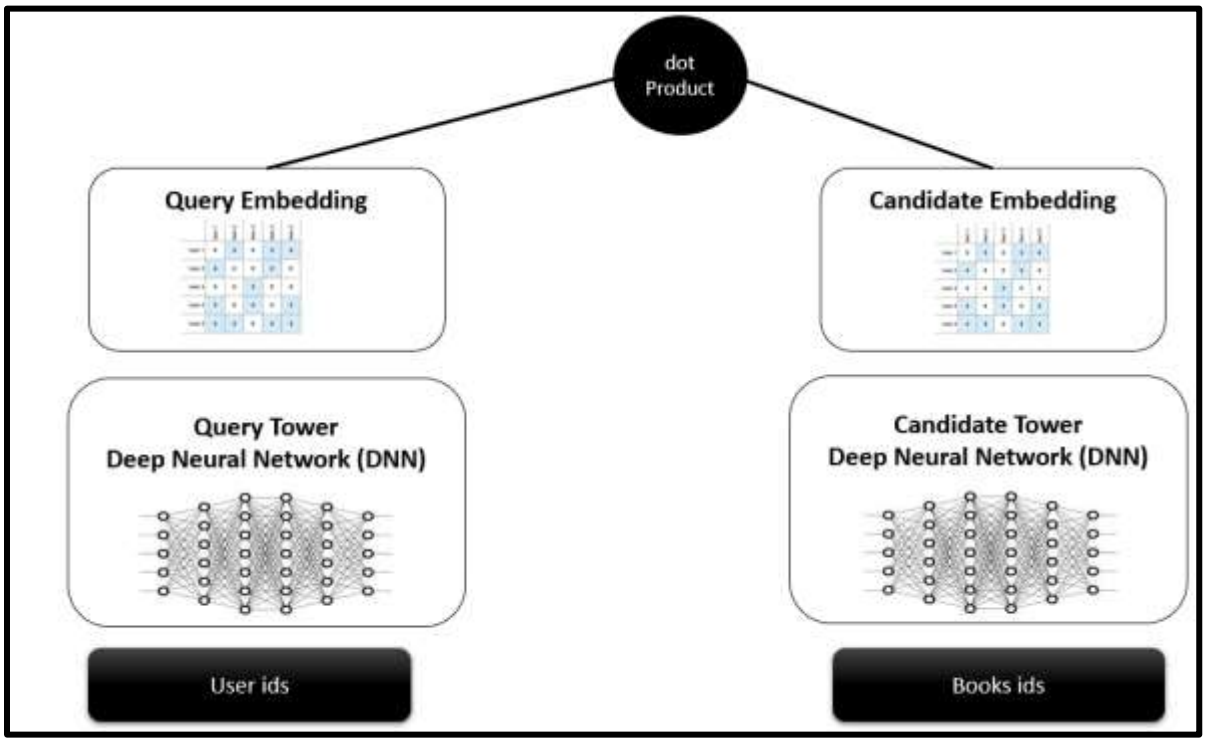

Figure 1: Two towers model

And for Ranking model :

In the ranking stage, the outputs of the retrieval model are taken and fine-tuned to select the best possible handful of recommendations. The task is to narrow down the set of items the user may be interested in, to a shortlist of likely candidates, typically numbering in the hundreds.

A model comprising multiple stacked dense layers is a relatively common choice for ranking tasks. (Figure 2) illustrates our recommendation model stages:

DOI:http://doi.org/10.24086/cocos2024/paper.1473

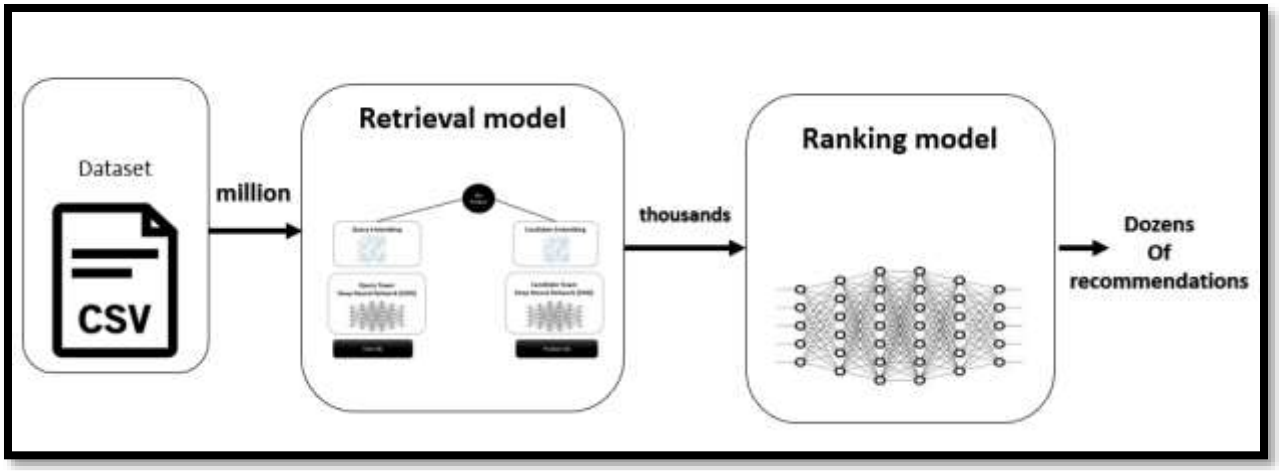

Figure 2: The Recommendation System with TensorFlow Recommenders

3. RESTful API Development

We have developed a user-friendly RESTful API using Node.js and express.js for smooth integration into mobile and web applications. This API acts as the connection point between our recommendation system and the Flutter framework, streamlining the integration process.

4. Flutter Framework

The Flutter framework, utilizing the Dart programming language, has been employed to manage the API endpoints created by Express.js, resulting in the development of a cross-platform application prototype. This application seamlessly integrates with the proposed recommendation system, providing end-users with an interactive experience.

5. Docker Containerization

We use Docker containers to deploy our recommendation models (Retrieval and Ranking) as a single docker image, to ensure compatibility across different platforms. Docker simplifies the deployment process and guarantees consistent performance across various operating systems and environments.

Figure 3, shows the architecture of our proposed model:

DOI:http://doi.org/10.24086/cocos2024/paper.1473

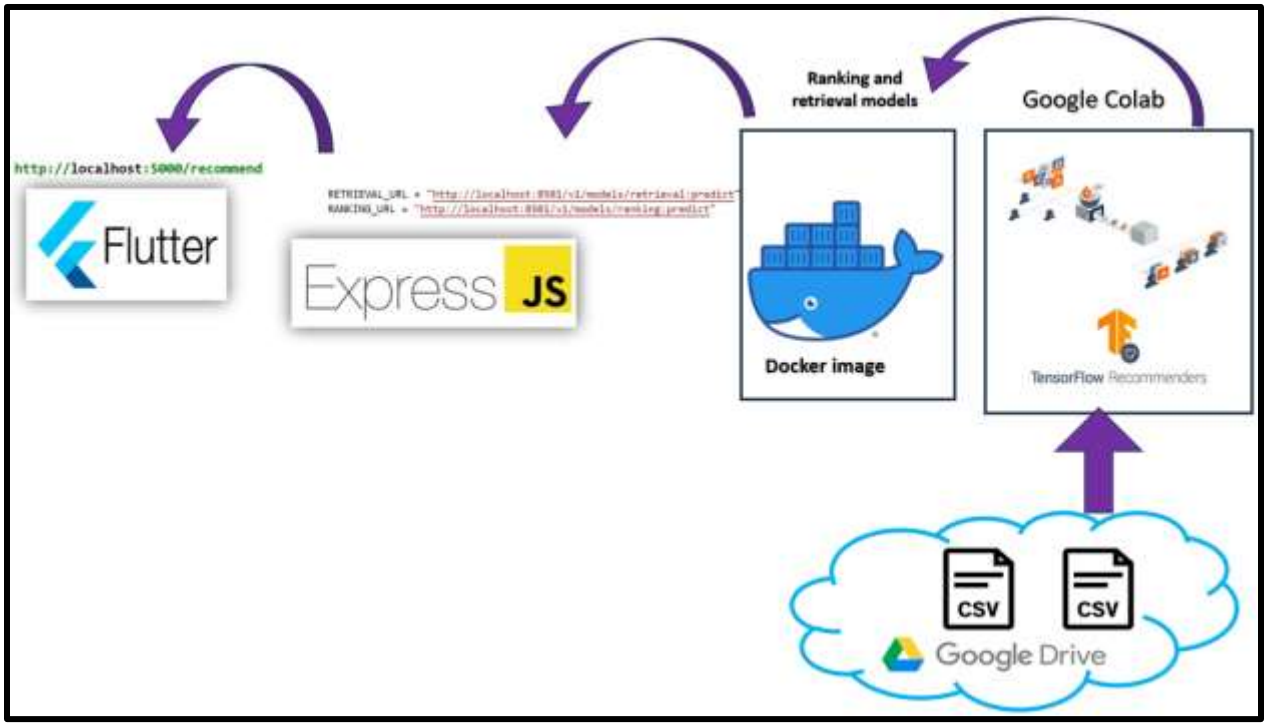

Figure 3: The Proposed model Architecture

IV. Result and discussion

In the process of creating recommendation models, we developed retrieval and ranking models.

In the Retrieval Model: This model used a Two-tower approach to create embeddings for both user IDs (query) and book titles (candidates). Its purpose was to identify similarities between users and books. By doing so, it helped reduce the dataset by considering only users and unique book titles, irrespective of user ratings. The evaluation results for this model are illustrated in figure 4:

```
{'factorized_top_k/top_1_categorical_accuracy': 0.054099999368190765,
 'factorized_top_k/top_5_categorical_accuracy': 0.06840000301599503,
 'factorized_top_k/top_10_categorical_accuracy': 0.07805000245571136,
 'factorized_top_k/top_50_categorical_accuracy': 0.1075500026345253,
 'factorized_top_k/top_100_categorical_accuracy': 0.12665000557899475,
 'loss': 29541.4140625,
 'regularization_loss': 0,
 'total_loss': 29541.4140625}
```

Figure 4: the result of retrieval model evaluation

DOI:http://doi.org/10.24086/cocos2024/paper.1473

The results show the following:

1. **Accuracy**:
   - Top-1 Accuracy: 5.41% of the time, the model correctly recommends the first-place item.
   - Top-5 Accuracy: 6.84% of the time, the model correctly recommends an item within the top 5.
   - Top-10 Accuracy: 7.81% of the time, the model correctly recommends an item within the top 10.
   - Top-50 Accuracy: 10.76% of the time, the model correctly recommends an item within the top 50.
   - Top-100 Accuracy: 12.67% of the time, the model correctly recommends an item within the top 100.
2. **Loss**: Total Loss: The overall prediction error is 29541.4140625, with no additional regularization applied.

In summary, the model exhibits relatively low top-1 accuracy but performs better in top-5 and top-10 recommendations. The high overall prediction error suggests potential for improvement through additional training or by exploring alternative model architectures.

And the ranking model sorts the recommendations generated by the retrieval model based on the user's ID, book title, and ratings data. This helps to filter out the less relevant recommendations and provide a shortlist of the most likely candidates, this is the result of evaluating this model figure 5:

```
{'root_mean_squared_error': 1.3426170349121094,
 'loss': 1.8060245513916016,
 'regularization_loss': 0,
 'total_loss': 1.8060245513916016}
```

Figure 5: the results of ranking model evaluation

The results show the following:

- **RMSE**: Measures how well the model's predictions match actual values, with lower values being better. RMSE here is approximately 1.34, meaning predictions are off by about 1.34 units on average.
- **Loss**: A general error measure; lower values indicate better performance. The loss is approximately 1.81 in this case.
- **Regularization Loss**: Reflects penalty for preventing overfitting. A value of 0 suggests no additional regularization.
- **Total Loss**: The overall model performance, including prediction error and regularization. It's approximately 1.81, indicating no extra regularization.

OR

the Root Mean Squared Error (RMSE) is 1.34, the loss is 1.81, the regularization loss is 0, and the total loss is 1.81. This indicates that the model is not performing very well. The RMSE is relatively high, and the total loss is also high. This suggests that the model is not able to

DOI:http://doi.org/10.24086/cocos2024/paper.1473

accurately predict the ranking of the items in the dataset.

After that the two models has been deployed as a TensorFlow saved model to the docker container as illustrated in the figure (6):

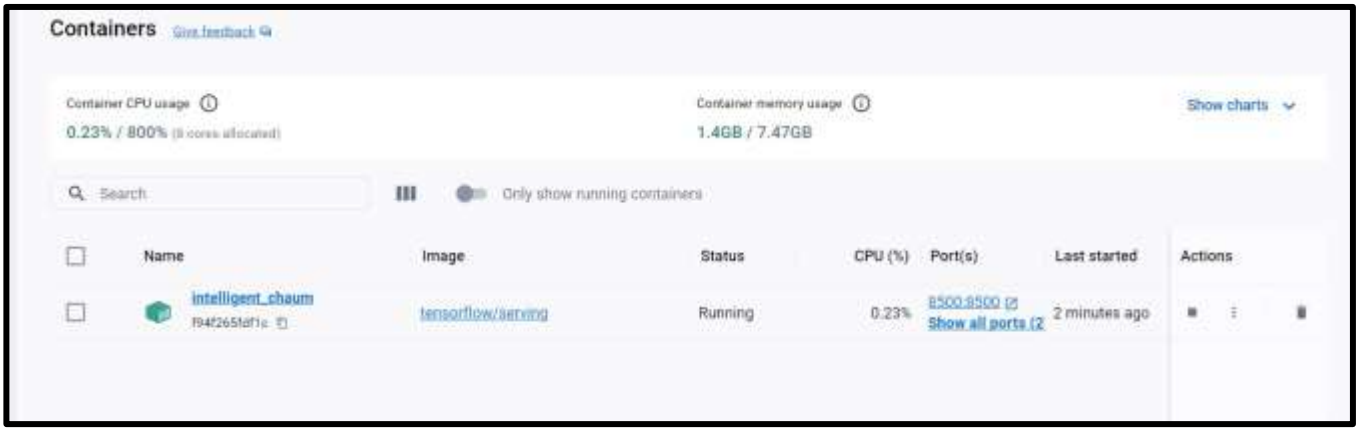

Figure 6: run TensorFlow models in the docker container.

and generate the restful API using express.js, to give a recommendation as an endpoint using post request with user_id, and respond with the recommendations,

finally creating a flutter application to make use of this API and make a cross-platform application like in figure (7):

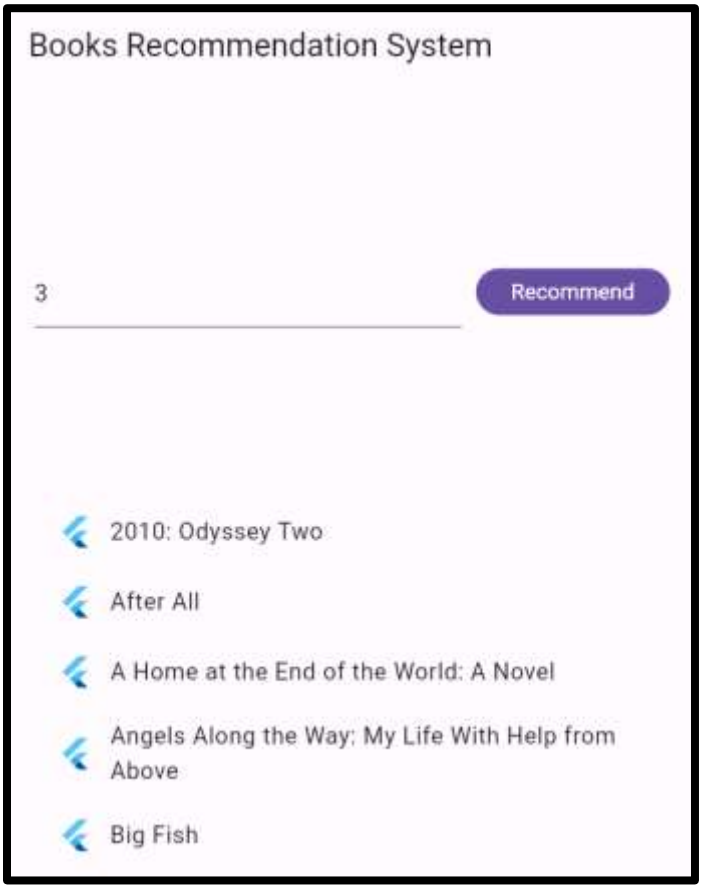

Figure 7: flutter recommendation application

DOI:http://doi.org/10.24086/cocos2024/paper.1473

V. Limitations:

Limited computer resources: We were only able to use a fraction of our dataset (100,000 examples) due to constraints in available RAM, CPU, and GPU resources, as well as a slow and unstable internet connection.

Suggestions for future improvement:

- Cloud computing resources: We can improve the performance and reliability of our recommendation system by leveraging cloud computing resources, such as Google Colab's premium offerings. This will provide us with access to more substantial computational power and storage capacity, which will allow us to train and evaluate our model more comprehensively.
- Larger dataset: Using a larger dataset can help improve model performance by capturing a broader range of user behaviors and preferences. This will lead to more accurate recommendations and evaluations.

VI. Conclusion

In the digital commerce era, recommendation systems are crucial for improving user experiences and boosting business growth. However, building and integrating these systems can be daunting, especially for developers lacking extensive machine learning knowledge. This research paper offers an innovative solution to simplify the development and deployment of mobile commerce-focused recommendation systems.

Our methodology leverages cutting-edge technologies like TensorFlow Recommenders, Docker, Node.js, and Flutter to create a robust and adaptable recommendation system. By encapsulating TensorFlow models in Docker containers and providing a user-friendly RESTful API, we enable effortless integration into mobile (Android and iOS) and web applications. Additionally, adopting the Flutter framework allows us to build a cross-platform recommendation system, ensuring compatibility.

This research delivers several key contributions. Firstly, it introduces a novel approach to streamline the development of mobile commerce recommendation systems, making it accessible to a broader developer audience. Secondly, it demonstrates the effectiveness of using TensorFlow Recommenders, Docker, and Node.js to construct recommendation systems with seamless integration capabilities via a user-friendly API. Finally, it provides a practical solution for creating cross-platform recommendation systems using the Flutter framework.

While our retrieval and ranking models show promise, there is room for improvement in terms of accuracy and prediction error. Future work may focus on optimizing these models and exploring alternative architectures for better performance.

To sum up, this research paper offers an integrated and practical solution to the complex challenge of developing and deploying recommendation systems for mobile commerce. By combining TensorFlow models with Docker, Node.js, and Flutter, it presents a versatile and user-friendly system, ensuring effortless communication and integration within mobile and web applications. This research paves the way for more accessible and effective recommendation systems in the dynamic landscape of mobile commerce, benefiting both developers and end-users.

VII. References

Akanferi, A. A., Asampana, I., Matey, A. H., & Tanye, H. A. (2022). ADOPTION OF MOBILE COMMERCE AND MOBILE PAYMENTS IN GHANA: AN EXAMINATION OF FACTORS INFLUENCING PUBLIC SERVANTS. *Interdisciplinary Journal of Information, Knowledge, and Management*, *17*, 287–313. https://doi.org/10.28945/4981

Anandaraj, A., Yeshwanth Ram, P., Sri Ram Kumar, K., Revanth, M., & Praveen, R. (2021). Book Recommendation System with TensorFlow. *2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021*, 1665–1669. https://doi.org/10.1109/ICACCS51430.2021.9441927

DOI:http://doi.org/10.24086/cocos2024/paper.1473

Bellini, P., Palesi, L. A. I., Nesi, P., & Pantaleo, G. (2023). Multi Clustering Recommendation System for Fashion Retail. *Multimedia Tools and Applications*, *82*(7), 9989–10016. https://doi.org/10.1007/s11042-021-11837-5

Filus, K., & Domańska, J. (2023). Software vulnerabilities in TensorFlow-based deep learning applications. *Computers and Security*, *124*. https://doi.org/10.1016/j.cose.2022.102948

Fudholi, D. H., Rani, S., Arifin, D. M., & Satyatama, M. R. (2021). Deep Learning-based Mobile Tourism Recommender System. *Scientific Journal of Informatics*, *8*(1), 111–118. https://doi.org/10.15294/sji.v8i1.29262

Guo, Y., Yin, C., Li, M., Ren, X., & Liu, P. (2018). Mobile e-commerce recommendation system based on multi-source information fusion for sustainable e-business. *Sustainability (Switzerland)*, *10*(1). https://doi.org/10.3390/su10010147

Kaggle Datasets. (n.d.).

Kamble, R. S., Shetty, S. D., & Jose, A. (2021). *APPLICATION DEVELOPMENT FOR MUSIC RECOMMENDATION SYSTEM USING DEEP DETERMINISTIC POLICY GRADIENT*.

Kantepe, E., Altikardes, Z. A., & Erdal, H. (2020, October 15). Product Recommendation System with Explicit Feedback Using Deep Learning Methods. *Proceedings - 2020 Innovations in Intelligent Systems and Applications Conference, ASYU 2020*. https://doi.org/10.1109/ASYU50717.2020.9259814

Lou, F. (2022). E-Commerce Recommendation Technology Based on Collaborative Filtering Algorithm and Mobile Cloud Computing. *Wireless Communications and Mobile Computing*, *2022*. https://doi.org/10.1155/2022/7321021

Lu, P., & Liu, P. (2023). Product Recommendation System Based on Deep Learning. *International Journal of Advanced Network, Monitoring and Controls*, *8*(1), 1–9. https://doi.org/10.2478/ijanmc-2023-0041

Tahir, M., Enam, R. N., & Mustafa, S. M. N. (2021). E-commerce platform based on Machine Learning Recommendation System. *IMTIC 2021 - 6th International Multi-Topic ICT Conference: AI Meets IoT: Towards Next Generation Digital Transformation*. https://doi.org/10.1109/IMTIC53841.2021.9719822

Venkatrama, S. (2017). A proposed business intelligent framework for recommender systems. *Informatics*, *4*(4). https://doi.org/10.3390/informatics4040040

Xu, Q., & Wang, J. (2022). A Social-aware and Mobile Computing-based E-Commerce Product Recommendation System. *Computational Intelligence and Neuroscience*, *2022*. https://doi.org/10.1155/2022/9501246

Zhu, W. (2022). Research on multi-source mobile commerce service recommendation model of data fusion based on tree network. *Concurrency and Computation: Practice and Experience*, *34*(13). https://doi.org/10.1002/cpe.5862

DOI:http://doi.org/10.24086/cocos2024/paper.1473