Developing Deep Learning Model for Cold Start Challenge in Recommendation Systems

Chien-Ting Lin
C.Lin42@lse.ac.uk
Candidate Number: 34462
London School of Economics and Political Science
London, UK

Dina Dai
Y.Dai28@lse.ac.uk
Candidate Number: 33328
London School of Economics and Political Science
London, UK

Jing Chen
J.Chen167@lse.ac.uk
Candidate Number: 35358
London School of Economics and Political Science
London, UK

Qinhan Zou*
Q.Zou2@lse.ac.uk
Candidate Number: 24668
London School of Economics and Political Science
London, UK

Final Project ST446, April 2024, London,UK

ABSTRACT

The field of Recommendation Systems (RS) has seen substantial growth, particularly in addressing the challenge of recommending items to new users with no interaction history, known as cold-start users. While numerous studies have explored recommendations for cold-start items, less attention has been given to cold-start users. Traditional methods like content-based filtering (CB) and collaborative filtering (CF) often fall short when it comes to new users due to the lack of historical interaction data. This study aims to address the gap by focusing specifically on entirely cold-start users within the context of restaurant recommendations. We developed a hybrid model that integrates Deep Neural Networks (DNN) with Alternating Least Squares (ALS) using Apache Spark, enhancing the ability to predict user preferences in the absence of rating history. This model leverages implicit data such as restaurant ratings and rating counts, which helps to refine the user-item matrix more effectively compared to using ALS alone. The integration of ALS and DNN in our hybrid model has shown promising results in overcoming the limitations of traditional RS approaches, particularly in handling cold-start scenarios. The hybrid model demonstrated superior performance, achieving a PR-AUC value of 0.89 on user cold start data and generally performing better on cold start than non-cold start data by a margin of 0.05 to 0.07. Overall, the hybrid model yielded better outcomes across datasets of both 10% and 20% compared to ALS only. This indicates that our model not only addresses the cold-start user problem effectively but also enhances overall recommendation quality in real-world scenarios where new users continuously enter the system.

KEYWORDS

Alternating Least Squares, Deep Neural Networks, Recommendation System, Distributed Computing, Cold Start

Statement:
Chien-Ting Lin, Dina Dai, Jing Chen, and Qinhan Zou. 2024. Developing Deep Learning Model for Cold Start Challenge in Recommendation Systems. Final Project for ST446, London, UK, 8 pages.

*All authors contributed equally to this research.

1 INTRODUCTION

Recently, the Recommendation Systems (RS) has become a popular research area and is applied to many areas such as commercial products, restaurants, books, research queries, and so on. Recommending cold-start items has been the fundamental challenge in the recommendation system, together with making recommendations to the cold-start users. This is because of the difficulty to involve accurate recommendations for new users or items with limited interaction histories. While the first issue has been discussed in many studies, there is less attention to the second one. Systems that employ content-based filtering (CB) or collaborative filtering (CF) solely are less effective when inferring user preference for cold-start users. To solve this problem, previous studies have developed various hybrid models, such as incorporating side information (i.e., content features) into the CF approach , or employing deep learning neural network techniques. However, in addition to the use of content features and the collaborative representations, the interplay between them has not been well explored, and thus leading to less efficient performance on the cold-start recommendation problem.

More importantly, even though many recent studies addressed the importance of dealing with cold-start problems, most of them focus on cold-start items but less attention of cold-start users. Generally, the cold-start problem stems from predicting the preferences for new items that have no or lack of user interactions in the training dataset, as well as new users who are new to the test dataset. In this study, we concentrate on making recommendations for ***entirely*** cold-start users that have no rating history, a task that is less explored in previous studies. The study focuses on exploring whether the use of Deep Neural Network (DNN) based on Alternating Least Squares (ALS) could potentially improve the predictions for cold-start users in the context of restaurant recommendations. We developed a hybrid model integrating ALS using Apache Spark and DNN to predict whether users will rate a restaurant with the highest score out of a five-star rating.

2 METHODOLOGY

2.1 Categories of Recommendation Systems

As electronic transactions via the Internet become more popular recently, there has been an increased interest in the development of recommendation systems. The aim of a recommendation system is to generate suggestions on products, such as movies, music, and books, that users will be more likely to purchase based on their interests or ratings by other users [5]. These systems use statistical analyses to calculate the probability that a user will buy a product at each location, ensuring that users get recommendations for the appropriate products to purchase. In addition to traditional products, recommendation systems are also applied to other fields, such as friend recommendation which is common in commercial or social platforms like instagram, and GP recommendation system used by NHS.

2.1.1 ***Content-based Filtering.*** Many algorithms have been applied to building these recommendation systems, with popular approaches including content-based filtering (CB) and collaborative filtering (CF) methodologies (Melville & Sindhwani, 2010). Usually, CB uses a set of items rated by an individual user and the characteristics of items themselves to infer a user profile, which is then applied to suggest other items that might be of interest [4]. In other words, CB will generate the content information of items, user profiles, and preferences based on features of the items (Figure 1). Specifically, the item profile includes multiple features of the items, both structured or unstructured. For example, in the case of restaurants, the item profile for each restaurant could be built based on various types of categories. A matrix will be created with restaurants as the rows and the categories as the columns, followed by binary representation, for example, to show whether the restaurant belongs to the category. In terms of the consumer profile, a preference matrix will also be built. Then, the restaurant content and the consumer profile will be compared to calculate the similarity between them.

The drawbacks of CB is that the identification of similarities in items based on the same attribute or characteristics would lead to highly specialised recommendations that are limited to items very similar to those already known to the user [6].

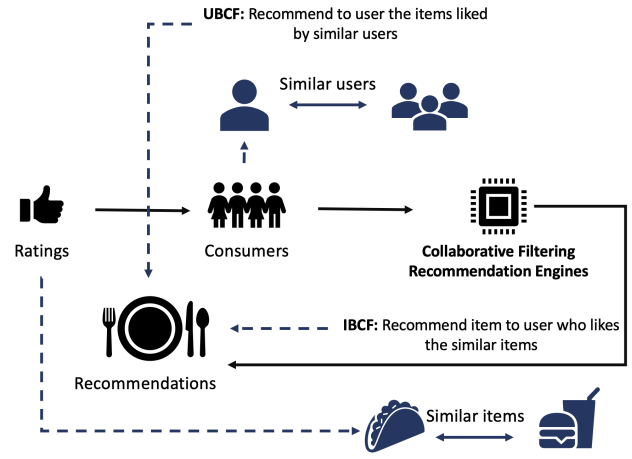
Figure 1: Content-based Recommendation System

2.1.2 ***Collaborative Filtering.*** Compared to CB, CF uses information filtering techniques to collect and analyse data based on user behaviours, activities, or preferences to predict what users will like, based on similarities between users [6]. CF will learn the quality representations of users and items based on the historical interactions, such as rates, clicks, ranks, etc., and therefore is successful in making personalised recommendations [3, 7]. More specifically, based on the ratings of a user, other users with similar preferences in the past will be found as the "neighbours"; then predictions will be made for all unknown products that the user has not rated, based on the preferences of their neighbours (Figure 2).

The main advantages of CF is the user-based collaborative filtering (UBCF) and item-based collaborative filtering (IBCF). The underlying assumption behind UBCF is that users are more likely to favour items liked by similar users. Similarly, IBCF suggests that by calculating the similarities between items, the unrated items could be recommended to the users. Therefore, the historical interactions between users and items are essential for creating high-quality collaborative embeddings, while the detailed information of items is not necessary. Recently, this approach incorporates more advanced Neural Network and Deep Learning approaches.

However, the main disadvantages of this approach are the sparsity problem and scalability problem [8]. In terms of the scalability problem, it is caused by the difficulty to effectively process and make recommendations as the number of users of items grows

Figure 2: Collaborative Filtering Recommendation System



rapidly. When CF is applied to large datasets, its performance may decrease significantly. Regarding the sparsity problem, this is because in real-world scenarios, many items may only receive few ratings, or many users rate only a small subset of available items, which leads to the situation that the user-item ratings matrix may be largely empty. As a result, it is difficult for the recommendation algorithm to find enough common ratings between users and to accurately predict preferences. Therefore, the sparsity of data leads to less reliable and less personalised recommendations because of the failure to identify connections between users and items.

While CB does not encounter cold-start problems, cold start problems are a key challenge for CF approach. As CF only requires the users who have interaction history with the system, the item information and the users' ratings, when CF is applied to the condition where there are lots of new items that do not have history interactions, the cold-start problem arises. Additionally, the cold-start problem also exists when recommending to new users who do not have rating information.

2.1.3 ***Hybrid Filtering*** . Given the drawbacks of CB and CF systems, the hybrid recommendation system is developed to use a combination of content-based and collaborative filtering methods to ensure that the recommended item is suitable. One common method is the feature combination approach, when a user's profile from CB is merged with user-item rating data to develop a hybrid approach. Another widely used method is via weighting. Initially, recommendation outcomes are combined, assigning equal weights to each result, which are then gradually modified based on user feedback to the recommendations. Overall, the hybrid system is the most widely used in reality because it has the potential to remove any defect that could take place when implementing a recommendation system[6] .

One hybrid recommendation model combines features of CF and deep learning neural networks to solve the complete cold start (CCS) problem where no rating records are available and incomplete cold start (ICS) problem [9]. Specifically, they use the deep neural network SADE to extract the content features of the items, followed by taking the content features into prediction of ratings for cold start items. Their experiments on Netflix movies ratings showed that tight coupling of CF approach and deep learning neural network is feasible and very effective for cold start item recommendation. It is indicated that the user experience and trust of the recommendation systems could be largely improved by solving the CCS problem and getting CCS items promoted. Therefore, the model could be generalised to other recommendation systems such as social networking or online shopping applications.

Nevertheless, it is also challenged that the use of RMSE rating prediction in this problem is not efficient enough in real world case.

In this project, we developed a hybrid model integrating Alternating Least Squares (ALS) and deep neural networks (DNN), which combine

2.2 Alternating Least Squares (ALS) & PySpark

Considering the potential risks of using CB or CF, new approaches of recommendation systems are developed to better solve real-world problems. Recently, matrix factorization techniques have been widely discovered. Compared to the more traditional algorithms, this method works by decomposing the user-item interaction matrix into lower-dimensional matrices, and thus simplifying the identification of latent factors that influence user preferences [**?**]. Gosh and their colleagues developed a recommendation system by using the algorithm for matrix factorization ALS (Alternating Least Square) in 2021. It is a type of CF technique used to tackle the over-fitting problem in sparse data [1].

Particularly, ALS model approximates the ratings matrix by factoring it into lower-dimensional user and product matrices. It optimises these factors to minimise the error between the actual and reconstructed observed ratings. Unknown ratings for items could then be predicted by multiplying these factors together.

Consequently, It could be applied in real-world circumstances to suggest products.

Given the advantage of ALS to deal with sparse data, the current project used it as the fundamental approach, calculating embeddings for users and items from user-item interaction data through matrix factorisation. These embeddings are used to infer user preferences and characteristics based on the user's rating patterns. In our cases, it was chosen to avoid over-fitting issues in the sparse google restaurant rating data, and hence increase the prediction accuracy.

However, because ALS is a type of CF model, it could not handle cold-start problems. This is because for new restaurants or users, there was no interaction history to make inferences. Therefore, we decided to train DNN based on the output of ALS.

2.3 Deep Neural Networks (DNNs)

Deep learning based recommendation systems became popular recently, which learned from both unstructured and unlabeled data, in order to make more intelligent decisions using artificial neural networks. Such neural networks could be decomposed into a user-item interaction matrix, which could be trained to predict the interactions based on both user and item features.

Deep Neural Networks (DNNs) include multiple hidden layers and nonlinearities, which is more powerful than shallow networks. However, it is difficult to train DNNs while the dataset is not large

enough. Therefore, we choose Google restaurant rating data, which is a sufficient dataset to train our hybrid model.

The current project chose to employ DNN for its advantages over traditional CF or CB targeting at cold-start problems (Zhang et al., 2019). Multi-Layer Perceptions (MLPs) enable more efficient feature representation and generate non-linear transformation between input data and output. Compared to the matrix factorisation used in ALS, DNN is more flexible to include various factors into modelling and creating different embeddings [2]. Therefore, in our project, we integrate deep learning embeddings for user and business IDs to provide a more detailed and dynamically adaptive representation of user and item characteristics based on the output from the ALS model. For users not encountered during the training phase, which are indicative of real-life cold-start scenarios, our model assigns zero vectors of 60 dimensions that correspond to the ALS rank. These zero vectors ensure consistency in the model's input structure and allow it to simulate the absence of prior interactions, thus effectively managing new user data. Importantly, the deep learning embeddings supplement the information for these users, enhancing the model's ability to handle users without existing data in the ALS features. This integration bridges gaps in user information, providing a robust framework for handling new and unseen user interactions. While the input for ALS only includes business ID, user ID and rating, DNN takes more indicators into account such as average rating, number of ratings per restaurant, and matrix of the interaction between user and restaurant, and so on.

As a result, by mapping user and business IDs to generate embeddings, we capture latent factors that extend beyond the explicit interaction data processed by ALS. These embeddings convert sparse, categorical inputs into dense, continuous vectors. The dynamic nature of these embeddings, learned during the training process, enhanced the model's ability to discern intricate patterns and adapt to new information, which is especially beneficial in environments where user preferences or item characteristics evolve over time. In real life, the model could be better applied to solve the problem faced by new restaurants or new users to the applications.

In addition to taking more information of the item, the use of DNN aims to improve the applicability of the hybrid model. Compared to the static ALS embeddings derived from direct user-item interactions, these deep learning embeddings offer a level of granularity and flexibility that allows continual learning and adjustment. This capability enables the model to refine predictions in response to shifting user behaviours and item details, thereby providing more personalised and accurate recommendations. The introduction of this advanced methodology significantly enhances the model's predictive accuracy and robustness, surpassing the capabilities of traditional ALS-based approaches.

To better deal with the cold-start problem, which is the main focus of our project, a critical innovation in our methodology is the strategic adjustment of cold-start instances in the training data to match those in the test dataset. This adjustment ensures that the model trains under conditions that closely mimic real-world deployment scenarios, which is essential for enhancing the model's predictive accuracy and reliability when dealing with new users or items that have sparse data.

We employ the calculation of the dot product between user and item features to capture the interaction relationship between users and restaurants, which is vital for assessing the strength and nature of these interactions. To align the cold-start proportions between the datasets, we first determined the ratio of zero dot products in both the training and testing sets. From these values, we derived an adjustment ratio, quantifying the extent to which the training set needed modification to mirror the distribution of cold-start scenarios found in the test set. Using this ratio, we randomly set a portion of the non-zero dot products to zero, artificially creating cold-start conditions within the training data. This manipulation was extended to the corresponding user_features and business_features, which were also set to zero at the selected indices to further replicate the test data's structure.

Furthermore, to address the class imbalance caused by the binary nature of our target variable (recommended/not recommended), we employed oversampling techniques for the minority class in the training dataset. Followed by the binary representation in CB approach as mentioned previously, we extended such binary representation to include the interaction between user and item. That is, indicators such as dot_is_zero, user_is_zero, and business_is_zero were incorporated to assist the model in recognizing and differentiating between genuine interactions and those adjusted to simulate cold-start conditions. These indicators play a vital role in enabling the model to accurately process and learn from both existing and synthetically adjusted data scenarios.

2.4 Model Procedure

The hybrid model procedure of our training process is shown by Figure 3.

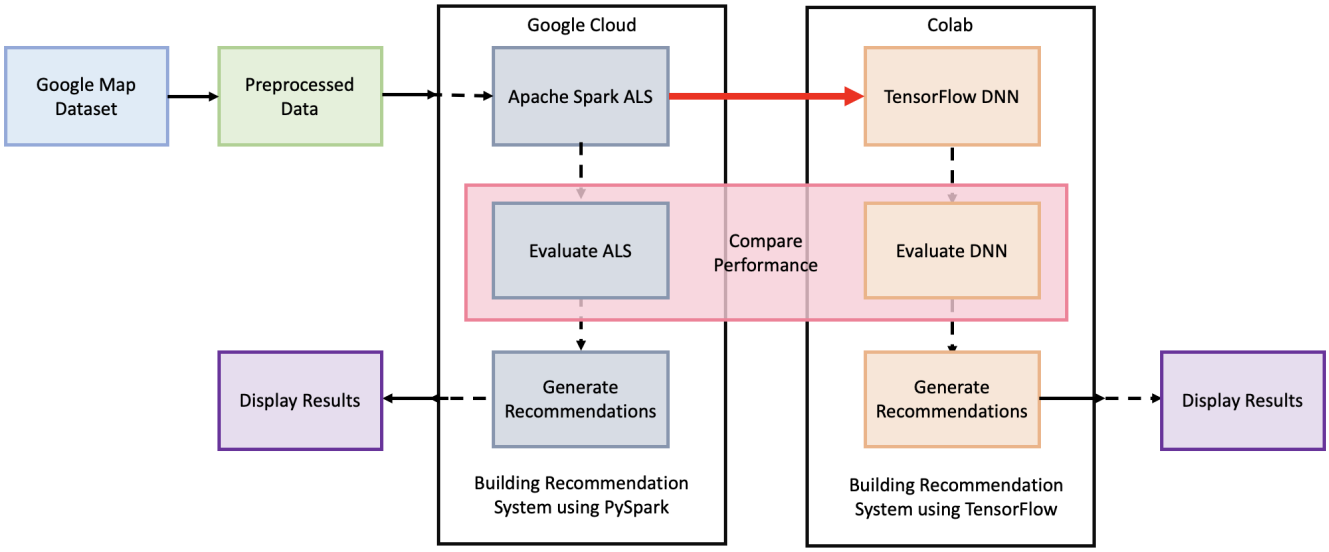
Input: Google Map Restaurant Dataset
Output: The probability of recommending a restaurant to a user
Step 1: Performing pre-process and exploratory analysis
Step 2: Pushing data to google cloud and setting up PySpark
Step 3: Training the ALS model over Hawaii dataset
Step 4: Analyses results of ALS
Step 5: Input the data and results from ALS using TensorFlow
Step 6: Train DNN model over Hawaii and California
Step 7: Analyse results from DNN, compared with ALS

2.5 Evaluation Criteria

During the evaluation phase, we use the test set to measure the model's efficacy. This involves integrating precomputed features with their corresponding entities in the test dataset. For users not encountered during the training phase, which are indicative of real-life cold-start scenarios, our model assigns zero vectors of 60 dimensions that correspond to the ALS rank. This setup ensures consistency in the model's input structure and allows it to simulate the absence of prior interactions, thus effectively managing new user data.

The evaluation metrics employed in this study include precision and recall, calculated by determining whether the system successfully predicts a recommendation for a given user and restaurant pair. Precision and recall vary depending on the chosen threshold, which can be adjusted based on the desired number of recommendations. Therefore, we use the precision-recall curve to capture the system's performance across all thresholds, assessing the predictive

Figure 3: Recommendation System Building Procedure



quality through metrics such as the area under the precision-recall curve and average precision.

Additionally, diversity in recommendations serves as a crucial evaluative metric in our study. By analyzing the diversity of the model's recommendations, we can observe its performance in suggesting a variety of restaurant options, particularly for cold-start users across different data volumes. This metric provides insights into how the model behaves when faced with limited user interaction data, which is a common scenario in real-world applications.

To evaluate the performance of our model, we utilised diversity as a critical metric to assess the breadth of the model's recommendations. To conduct a focused and meaningful analysis, we initially selected 1% of users from the test set, to evaluate the diversity of recommendations. For each user, we identified the top 10 restaurants based on the highest predicted ratings from our hybrid model. The diversity of these recommendations was then calculated by measuring the cosine similarity among the features of the recommended restaurants, with diversity quantified as (1 - average of these similarity scores). This approach highlights the model's ability to offer a variety of choices. A lower average score of similarity among top recommendations indicates greater diversity, demonstrating that the recommended businesses differ significantly in their features. By analysing diversity, we gain precise insights into how the model's recommendation strategy adapts to datasets of varying sizes.

3 DATA

3.1 Strategy of Data Processing

To address the cold-start problem in ALS model, we used Google Local Data (2021) [1], which encompasses review details from Google Maps up to September 2021 in the United States. This dataset includes rating data, such as ratings and timestamps, alongside business metadata, like addresses, review counts, and categories. The database predominantly comprises users with limited historical data where most users have provided only one rating. This characteristic closely aligns with our research question and mirrors the real-world scenario where the platform frequently needs to generate recommendations for new users without knowing their preferences. With its extensive data volume of over 600 million ratings from 100 million users, this database serves as an appropriate choice for the project, providing a comprehensive and realistic representation of the challenges associated with recommendation systems.

To deal with the extensive data, we employed distributed computing and Parquet for data storage and processing. Parquet, seamlessly integrated with Apache Spark, leverages partitioned and columnar storage to enhance the performance of data query and retrieval effectively.

We used the dataset of Hawaii because it represents the smallest data volume in the database, making it an ideal initial testing ground. It comprises 517,949 users and 12,769 operational restaurants. To understand the minimal amount of data necessary to achieve satisfactory results while efficiently utilising resources, we segmented this dataset into progressively larger sizes—10%, 20%, 50%, 80%, and 100%. This segmentation was strategically designed to allow us to methodically determine the smallest subset size that could provide reliable outcomes as well as minimising resource consumption and optimising data usage. Each subset was further divided, allocating 90% for training and 10% for testing. Given the dataset's considerable size, using only a 10% sample for testing suffices to produce statistically significant results.

This approach enabled us to assess the model's performance across varying data volumes and understand its capabilities and limitations in different data availability scenarios. Ultimately, after

[1] Google Local Data. https://datarepo.eng.ucsd.edu/mcauley_group/gdrive/googlelocal/.

verifying the model's performance on smaller subsets, we plan to apply it to the largest dataset in our collection, from California, to demonstrate its efficacy and scalability in a significantly more extensive and diverse environment.

3.2 Exploratory Data Analysis

We gained insights into the data of Hawaii through visualization.

Figure 4 showed that most restaurants exhibit an average rating over 4, thus we regarded a user rating below 4 for a restaurant as not recommended and a rating over 4 as recommendation for modelling.

Figure 4: Distribution of Ratings

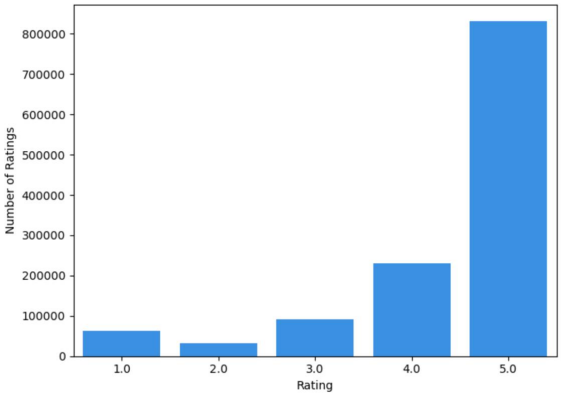
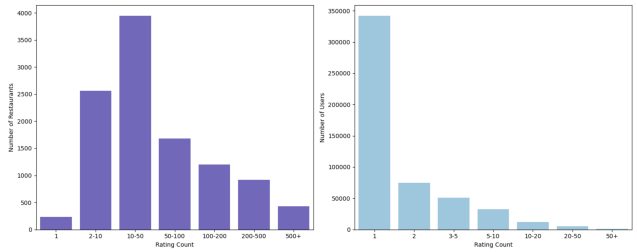


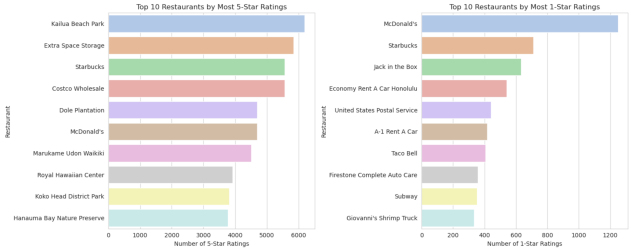
Figure 5 reflected the distribution of rating numbers received by each restaurant and given by each user respectively. Restaurants garnering 10-50 ratings are the most numerous with nearly 4,000 establishments falling into this category, while almost 350,000 users provided only one rating. Thus, nearly 67% users were considered as cold-start users.

Figure 5: Distribution of Rating Counts



Furthermore, the discovery from the Figure 6 that chain restaurants like Starbucks and McDonald's ranked highest both in receiving the most five-star ratings and the most one-star ratings underscores the importance of considering user diversity and preference complexity. To address this, additional features can be incorporated to offer personalised recommendations, especially for new users and recently launched restaurants.

Figure 6: Top 10 Restaurants by Most 5-Star / 1-Star Ratings



4 MODELLING AND EVALUATION

In the landscape of recommendation systems, our comparative study presents a numerical evaluation of two distinct approaches: the well-established Alternating Least Squares (ALS) model implemented in PySpark and a cutting-edge hybrid model that enhances ALS with deep neural networks (DNN). The ALS model, foundational to our exploration, was calibrated with precision to ascertain its utmost performance, especially in the context of smaller data subsets.

4.1 ALS Model Optimization and Configuration

Our initial foray used the ALS model, where we meticulously tuned the pivotal parameters of rank, maxIter, and regParam. The rank parameter, critical for capturing the data's complexity, was varied between 30 to 70. The maxIter parameter, which dictates convergence potential, and the regParam, a safeguard against overfitting, were also rigorously tested. Through a grid search analysis on the Hawaiian dataset, we determined that a rank of 60, a regularization parameter of 0.1, and 20 maximum iterations struck the optimal balance, minimizing the root mean square error (RMSE). These parameters were chosen after testing 30 different model configurations and were maintained for larger datasets to manage computational resources effectively. Due to computational resource constraints, we abstained from performing a grid search on the larger 80% and 100% California data subsets. Instead, we applied the optimal parameters gleaned from the smaller dataset, positing that these parameters would remain robust across larger volumes of data (as shown by Figure 7).

4.2 Numerical Assessment in Smaller Data Sets (Hawaii)

In the focused assessment of the smaller Hawaiian dataset, for the cold start users, the hybrid model, which integrates deep neural networks (DNN) with Alternating Least Squares (ALS), exhibited exceptional efficacy. A standout feature of this hybrid model is its remarkable performance using only a minimal portion of the dataset. From Figure 8, with just 10% of the data, the hybrid model not only achieved an average precision of 0.80 but also demonstrated a significant diversity score of 0.93. These results underscore the model's capability to deliver robust and varied recommendations from a substantially reduced dataset size, highlighting its efficiency in learning and prediction.

Figure 7: ALS Precision-Recall Curves

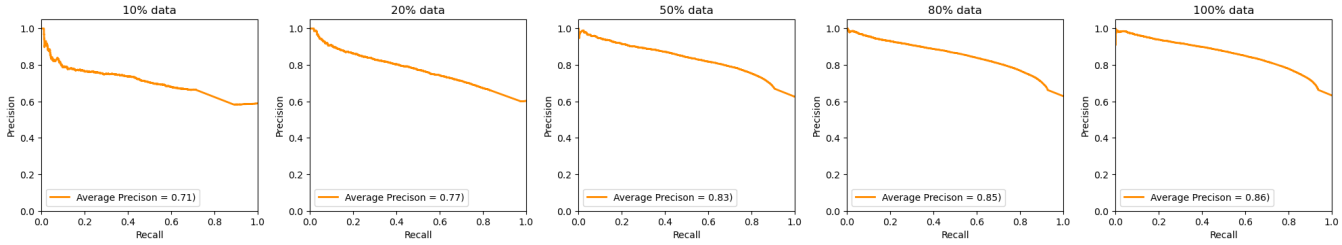
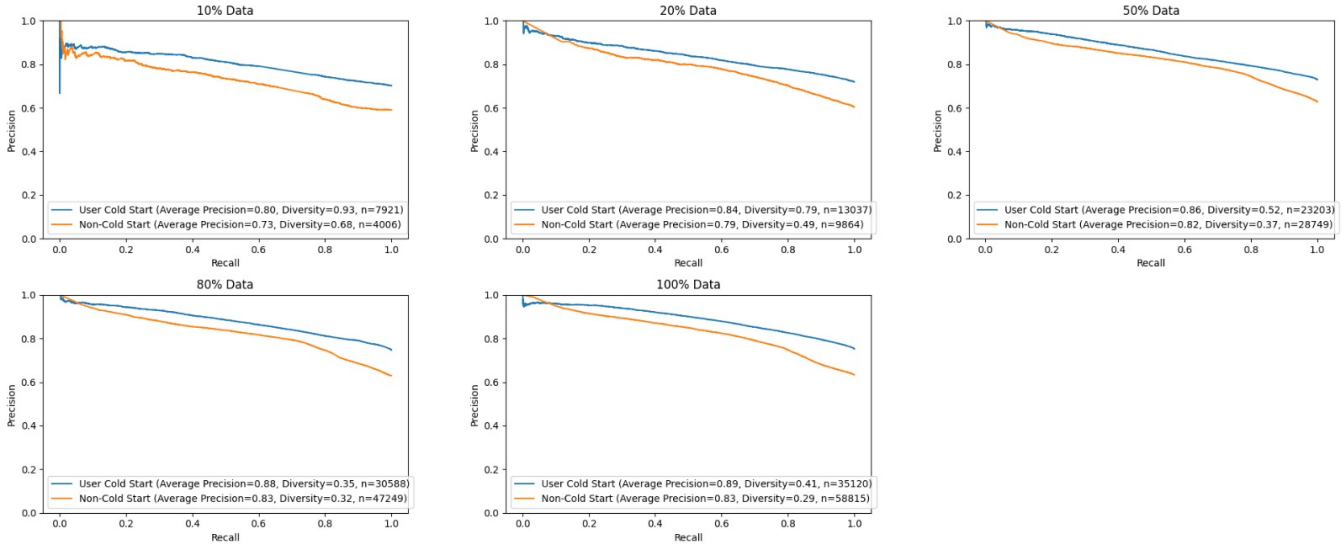


Figure 8: DNNs Precision-Recall Curves



We expanded the data volume to 20%, the hybrid model continued to enhance its average precision, reaching 0.84, while maintaining a high diversity score of 0.79. This upward trend in precision was sustained as the dataset increased; at 50% data volume, precision reached 0.88 with a diversity score of 0.52, and at 80%, precision rose further to 0.89, with diversity slightly increased to 0.35. By the time we evaluated the entire dataset (100%), the hybrid model's precision peaked at 0.89, and the diversity score increased to 0.41.

4.3 Diversity Trends and Data Volume in the Hybrid Model

This decline in diversity as data volumes increase offers a critical insight into the model's operational dynamics. When the model was applied to the full Hawaiian dataset, the diversity score notably dipped to 0.39. This trend indicates that as the dataset size grows, the model's recommendations tend to converge towards a narrower set of popular restaurants. Despite this reduction in diversity, such a strategy remains pragmatic, especially for cold-start users for whom the model's ability to recommend popular and broadly appealing options is crucial. This approach showcases the model's practical utility by balancing the need for precision with the attractiveness of the recommendations.

4.4 Large Dataset Evaluation (California)

For the extensive California dataset, our approach strategically utilized only 10% of the data, a decision informed by the promising results from the smaller Hawaiian dataset at similar data volumes. This focused analysis allowed us to demonstrate the model's efficiency and effectiveness without the need for extensive computational resources. Within this larger dataset, the hybrid model not only achieved an average precision of 0.80 but also showcased exceptional performance in cold-start scenarios, a critical area where traditional ALS typically underperforms.

Remarkably, the hybrid model's performance in these scenarios not only excelled in comparison to its own performance across different dataset sizes but also surpassed the traditional ALS model's precision in non-cold start conditions. Specifically, as shown by Figure 9&10, while the ALS model achieved a precision of 0.78 in its ideal non-cold start scenarios within the same dataset, the hybrid model's precision in addressing cold-start challenges was higher, at 0.80. Moreover, the hybrid model achieved 0.98 in diversity in user cold start scenarios. This diversity metric is particularly remarkable, as it signifies the model's capacity to recommend a broad array of restaurants to new users, thereby ensuring a rich and varied selection that can cater to wide-ranging tastes and preferences.

Figure 9: ALS performance on CA data

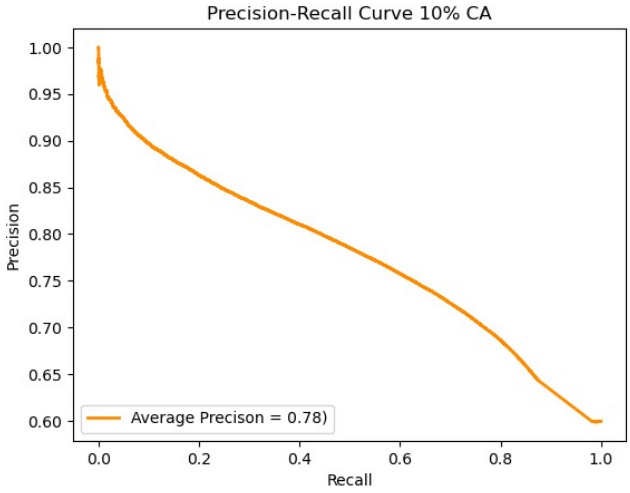
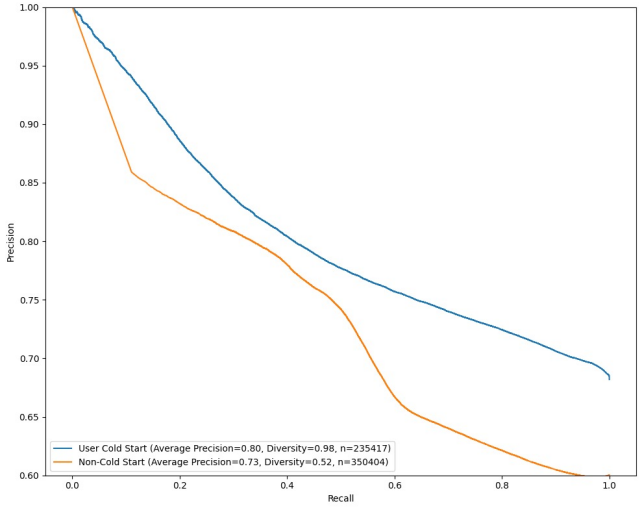


Figure 10: DNNs performance on CA data



5 CONCLUSION AND DISCUSSION

To sum up, the hybrid model combining ALS and DNNs tackled the cold start problem through two approaches. Firstly, it leveraged implicit data like the restaurant's average rating and rating count to improve the model's prediction performance. Secondly, it enhanced the relationship matrix between users and items effectively. While ALS also generated recommendations based on the user-item interaction, the sparse nature of its relationships limited its effectiveness in the cold start problem. Therefore, the hybrid model we developed effectively addressed the issue of ALS unavailable to predict on cold start data and achieved a PR-AUC value of 0.89 on user cold start data. Since the model is specifically trained for cold start problems, it performed slightly better on cold start data than on non-cold start data, with a gap around 0.05 to 0.07. Yet, it still outperformed ALS across datasets of both 10% and 20% sizes with diverse data types. This suggests that this model can effectively replace ALS when only a small portion of the data is utilised. Despite the diversity of recommendation outcomes decreased with the increase of data volume, this approach remains practical, particularly for new users who rely on the model's capacity to suggest popular and widely appealing choices.

Furthermore, this project has the following limitations. Due to the constraint of computational power, we conducted a limited range of grid search, particularly employing only one set of parameters for larger datasets of 80% and 100%. If more computational resources were available, the model's performance could be better. Moreover, while this model was specifically trained for cold start data, developing a separate model for non-cold start data and subsequently combining the two models could yield a recommendation system applicable to more scenarios. The project didn't consider temporal changes or real-time information, thus was unable to capture trends in user preferences over time. Addressing these shortcomings in future research could lead to the development of a more robust recommendation system capable of incorporating up-to-date data.

ACKNOWLEDGMENTS

This paper is the final project of ST446: Distributed Computing for Big Data, 2024. All authors contributed equally to this research.

REFERENCES

[1] Anas Alzogbi, Polina Koleva, and Georg Lausen. 2019. Towards Distributed Multi-model Learning on Apache Spark for Model-Based Recommender. In *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, Macao, Macao, 193–200. https://doi.org/10.1109/ICDEW.2019.00-12

[2] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, Boston Massachusetts USA, 191–198. https://doi.org/10.1145/2959100.2959190

[3] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. (2020). https://doi.org/10.48550/ARXIV.2002.02126 Publisher: [object Object] Version Number: 4.

[4] Umair Javed, Kamran Shaukat, Ibrahim A. Hameed, Farhat Iqbal, Talha Mahboob Alam, and Suhuai Luo. 2021. A Review of Content-Based and Context-Based Recommendation Systems. *International Journal of Emerging Technologies in Learning (iJET)* 16, 03 (Feb. 2021), 274. https://doi.org/10.3991/ijet.v16i03.18851

[5] Heung-Nam Kim, Ae-Ttie Ji, Inay Ha, and Geun-Sik Jo. 2010. Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. *Electronic Commerce Research and Applications* 9, 1 (Jan. 2010), 73–83. https://doi.org/10.1016/j.elerap.2009.08.004

[6] Prem Melville and Vikas Sindhwani. 2017. Recommender Systems. In *Encyclopedia of Machine Learning and Data Mining*, Claude Sammut and Geoffrey I. Webb (Eds.). Springer US, Boston, MA, 1056–1066. https://doi.org/10.1007/978-1-4899-7687-1_964

[7] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. (2012). https://doi.org/10.48550/ARXIV.1205.2618 Publisher: [object Object] Version Number: 1.

[8] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2000. *Application of Dimensionality Reduction in Recommender System - A Case Study:*. Technical Report. Defense Technical Information Center, Fort Belvoir, VA. https://doi.org/10.21236/ADA439541

[9] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2016. Collaborative Filtering and Deep Learning Based Hybrid Recommendation for Cold Start Problem. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*. IEEE, Auckland, 874–877. https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2016.149