# Flipkart Grid 5.0

## Personalized Product Recommendation System

### Objective

The goal is to enhance user experience by implementing a personalized product ranking system. The task involves designing and implementing a solution that predicts the most suitable products for users based on their unique characteristics and preferences.

### Data

We started by generating synthetic data that included user profiles, product categories, and interaction patterns. The datasets were:

Users: This dataset contained user attributes like age, gender, location, and interests.

Products: Included product attributes like name, category, price, and average rating.

Interactions: Recorded user-product interactions, which primarily consisted of user ratings for products.

### Methodology

Various models were trained and evaluated using the generated data:

1. Neural Collaborative Filtering (NCF)

    Introduction: NCF is a deep learning-based recommendation algorithm that merges generalized matrix factorization and multi-layer perceptron for the task of collaborative filtering.

    How it helps in Recommendation Systems: NCF can capture intricate structures in the user-item interaction matrix by leveraging deep neural networks. It can make use of additional features, providing a more holistic recommendation.

2. Linear Regression

    Introduction: A simple yet effective machine learning model where the outcome variable is predicted based on one or more independent variables.

    How it helps in Recommendation Systems: It can provide a baseline performance and can be used to understand the relationship between different features and the target variable (user ratings in this case).

3. Random Forest

Introduction: An ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes for classification or mean prediction for regression.

How it helps in Recommendation Systems: Random Forest can capture non-linear relationships in the data and can handle large datasets with higher dimensionality. It can also handle missing values and maintain accuracy for missing data.

4. XGBoost

Introduction: An optimized gradient boosting algorithm known for its speed and performance.

How it helps in Recommendation Systems: XGBoost is known for its high efficiency, flexibility, and ability to handle missing values. It can capture non-linear relationships and is resistant to overfitting.

5. Deep Matrix Factorization (DeepMF)

Introduction: A hybrid model that combines matrix factorization and deep learning to predict user-item interactions.

How it helps in Recommendation Systems: DeepMF can capture both linear and non-linear relationships in the data. It can also leverage additional features, providing more personalized recommendations.

| | RMSE | MAE | Precision | Recall | F1 Score | Explained Variance |
|---|---|---|---|---|---|---|
| **NCF** | **0.4648** | **0.3835** | **NA** | **NA** | **NA** | **NA** |
| **Linear Regression** | 0.9848 | 0.6756 | 0.7957 | 0.8605 | 0.8268 | 0.3830 |
| **Random Forest** | 0.8517 | 0.4276 | 0.7945 | 0.8516 | 0.8221 | 0.5385 |
| **XGBoost** | 0.8219 | 0.4285 | 0.8174 | 0.8321 | 0.8247 | 0.5702 |
| **DeepMF** | 1.4174 | 1.1808 | 0.4731 | 0.4774 | 0.4752 | -0.2782 |

## Result

The Neural Collaborative Filtering (NCF) performed the best among all the other models.

## Challenges & Limitations

### Data Challenges:

- Synthetic Data: We generated synthetic data for users, products, and interactions. While this allows for flexibility, it may not truly reflect real-world user behaviors or preferences, leading to potential inaccuracies in predictions.

- Cold Start Problem: New users or products that haven't had any interactions can pose a challenge. Our models, especially collaborative filtering ones like NCF, might struggle to provide accurate recommendations for them.

## Model-Specific Challenges:

- NCF Limitations: Despite its superior RMSE and MAE scores, NCF's precision and recall were not satisfactory. This indicates that while the model's overall error might be low, it might not be capturing all potentially relevant recommendations.

- Linear Regression Assumptions: Linear regression assumes a linear relationship between features and the target variable. This might not always be the case in recommendation systems.

- Tree-based Models Overfitting: Models like Random Forest and XGBoost, if not properly regularized, can overfit to the training data, reducing their generalization to unseen data.

- DeepMF Complexity: DeepMF, being a deep learning model, requires more computational resources and can be more complex to tune and interpret compared to traditional machine learning models.

## General Challenges:

- Scalability: As the dataset grows, some models might become computationally intensive. Ensuring that the recommendation system scales efficiently with increasing data is crucial.

- Changing User Preferences: User preferences can change over time. A static model might not capture these evolving trends. Periodic retraining or adaptive models are essential.

- Diversity vs. Accuracy Trade-off: Often, there's a trade-off between providing diverse recommendations and highly accurate ones. Too much focus on accuracy might lead to a filter bubble, where users are only shown items very similar to their past preferences.

## Ethical and Bias Concerns:

- Bias in Recommendations: If the training data has inherent biases, the model might propagate or even amplify these biases in its recommendations.

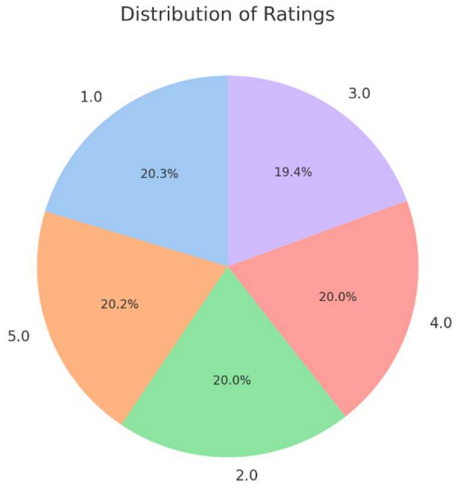
- Privacy Concerns: Collecting and using user data for personalizing recommendations raises privacy concerns. Ensuring user data is anonymized and used ethically is paramount.

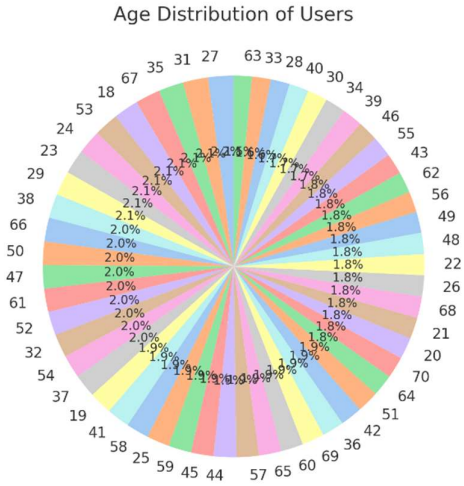
## Conclusion

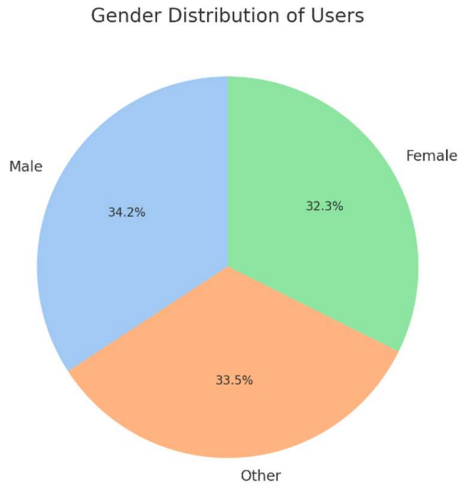
The models were evaluated based on RMSE, MAE, Precision, Recall, F1 Score, and Explained Variance. After thorough analysis and comparison, the Neural Collaborative Filtering (NCF) model stood out as the best-performing model. It yielded the lowest RMSE and MAE values, making it highly effective for predicting user preferences. XGBoost and other models also demonstrated significant potential, but NCF's superior predictive accuracy made it the most suitable model for our dataset. As always, when implementing such systems in real-world scenarios, it's essential to consider additional factors such as scalability, interpretability, and training time to ensure optimal performance and user satisfaction.

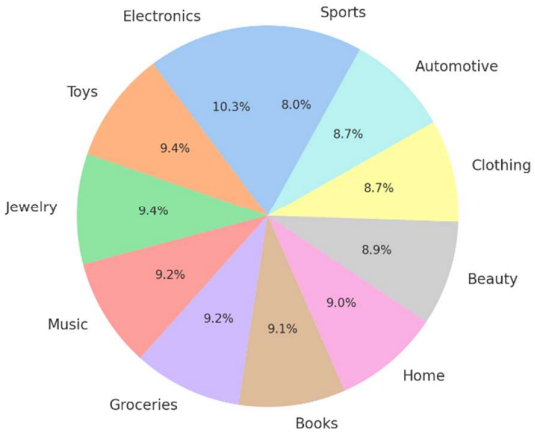
## Data Visualization

Three CSV files were generated with synthetic data for training the models. Here is the pictorial visualization of the data we had.



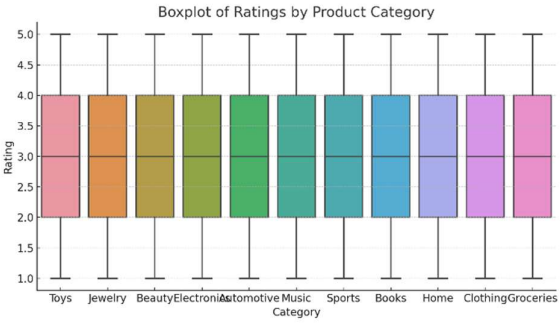



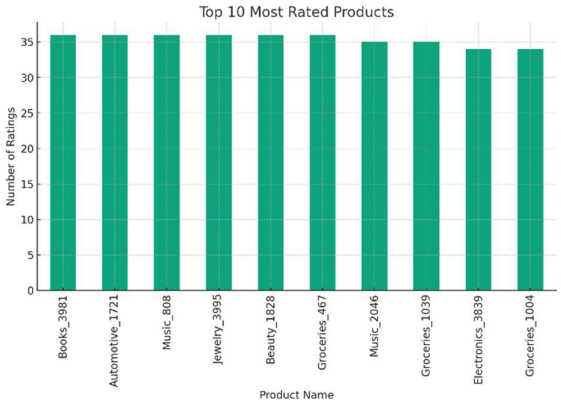



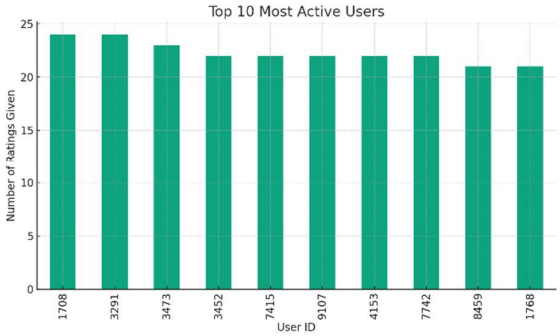



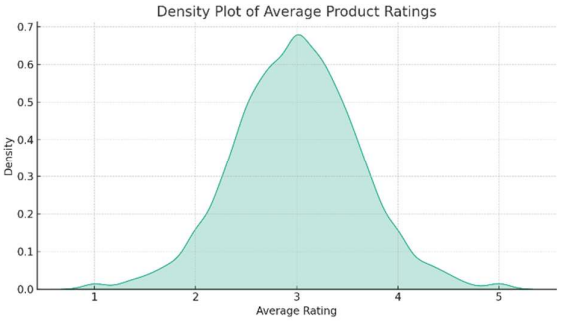
Here are a few more visualizations that might be of interest:

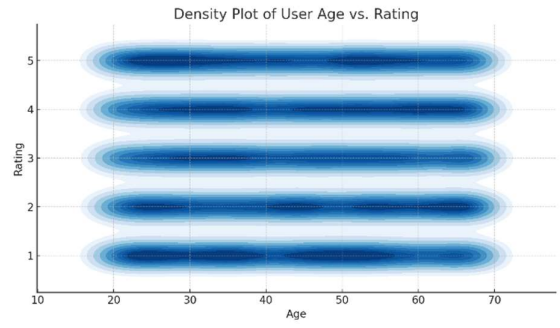
1. **Boxplot of ratings** for different product categories to understand the spread and central tendency of ratings per category.
2. **Bar chart** of the top 10 most rated products.
3. **Bar chart** of the top 10 most active users (users who gave the most ratings).
4. **Density plot (KDE)** of product average ratings to see the distribution of average product ratings.
5. **Density plot (KDE)** of user age against rating to see if there's any trend between age and the ratings provided.



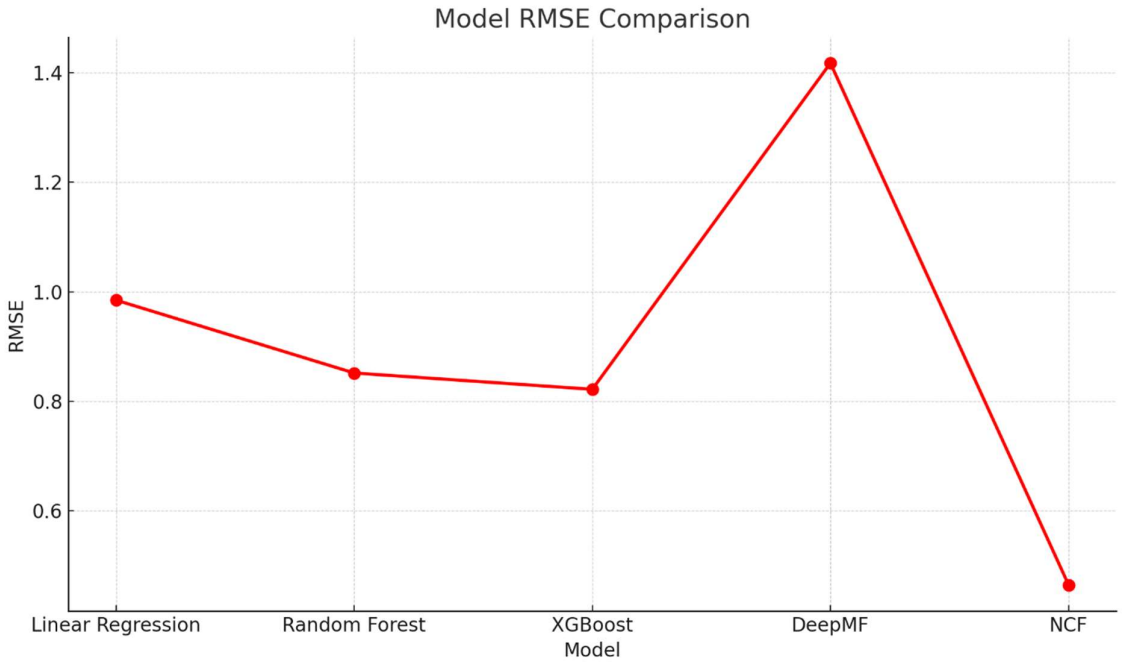



Here's a line chart comparing the RMSE of the different models:




From the chart, it's evident that the NCF model has the lowest RMSE, which implies the highest accuracy in terms of prediction error. Following closely is the XGBoost model, then Random Forest, Linear Regression, and finally DeepMF with the highest RMSE.