**Predicting User Purchase History with Deep Learning**

**Introduction**

**Objective**

The primary objective of this project is to develop a deep learning model that predicts the next purchase a user might make based on their previous purchase history. This model aims to enhance the recommendation system of GenZDealZ.ai by providing personalized recommendations to users.

**Background**

GenZDealZ.ai is an e-commerce platform that aggregates deals from various online retailers such as Amazon, Flipkart, and Myntra. The platform aims to improve its recommendation system by leveraging user purchase history data to predict future purchases.

**Data Simulation**

**Approach**

Due to the unavailability of real user purchase history data, we simulated the data to mimic real user behavior. This involved creating a dataset where each user has a sequence of purchases from different platforms.

**Data Example**

The simulated data is structured as follows:

python

Copy code

data = [

{'user': 'user1', 'purchases': ['amazon', 'flipkart', 'myntra']},

{'user': 'user2', 'purchases': ['amazon', 'flipkart']},

# More user data...

]

**Data Generation**

We used a Jupyter notebook (data\_simulation.ipynb) to generate the data and save it to a JSON file:

python

Copy code

save\_data\_to\_file(simulated\_data, 'simulated\_purchase\_history.json')

print(f"Generated data for {num\_users} users and saved to 'simulated\_purchase\_history.json'.")

**Data Preprocessing**

**Steps**

1. **Load Data**: Read the simulated purchase history data from the JSON file.
2. **Extract Purchase Sequences**: Extract sequences of purchases for each user.
3. **Tokenization and Encoding**: Tokenize the purchase sequences and convert them into sequences of integers.
4. **Padding**: Pad the sequences to ensure uniform length.

**Model Architecture**

**Overview**

We implemented a Recurrent Neural Network (RNN) model using TensorFlow's Keras API. The model includes embedding layers, SimpleRNN layers, dropout layers for regularization, and a dense layer for output prediction.

**Layers**

1. **Embedding Layer**: Transforms input sequences into dense vectors of fixed size.
2. **SimpleRNN Layers**: Captures temporal dependencies in the purchase sequences.
3. **Dropout Layers**: Prevents overfitting by randomly setting a fraction of input units to 0 during training.
4. **Dense Layer**: Outputs the probability distribution over the next possible purchase.

**Training and Evaluation**

**Data Preparation**

We prepared the predictors and label sequences and split the data into training and testing sets:

python

Copy code

# Preparing predictors and label sequences

X = padded\_sequences[:, :-1]

y = padded\_sequences[:, -1]

# Splitting the data

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

**Training**

The model was trained using the sparse\_categorical\_crossentropy loss function and the adam optimizer over 100 epochs with a batch size of 32:

**Evaluation Metrics**

We evaluated the model's performance using accuracy and loss metrics on both training and validation datasets.

**Results**

**Performance**

* **Training Accuracy**: The model achieved high accuracy on the training data.
* **Validation Accuracy**: The validation accuracy was slightly lower, indicating a potential overfitting issue.

**Findings**

* The model effectively captures the temporal dependencies in user purchase sequences.
* Dropout layers helped mitigate overfitting to some extent.

**Challenges and Solutions**

**Challenges**

1. **Data Simulation**: Ensuring that the simulated data realistically represents user behavior was challenging.
2. **Model Overfitting**: Balancing model complexity and generalization was difficult, leading to slight overfitting.

**Solutions**

1. **Realistic Simulation**: We aimed to make the simulated data as realistic as possible by considering typical user behavior patterns.
2. **Regularization**: Added dropout layers to prevent overfitting and improve generalization.

**Conclusion and Future Work**

**Conclusion**

The developed RNN model shows promising results in predicting user purchases based on historical data. It can potentially enhance the recommendation system of GenZDealZ.ai by providing personalized recommendations.

**Future Work**

* **Use Real Data**: Incorporate real user purchase history data for training and validation.
* **Advanced Architectures**: Experiment with more sophisticated models like LSTM or GRU for better performance.
* **Integration**: Integrate the model into the GenZDealZ.ai platform and evaluate its impact on user engagement.