Brief Business Problem :

100 – 500 char

Build a product recommendation system that is also capable of detecting the bias in recommendation, if any. We have two data sets, one for the electronic products and one for electronic products.

Proposed solution :

100 – 1000 char

Build a system that can predict rating given user ID, product ID and other features like model attribute etc., then we can use that rating to recommend item based on a threshold value e.g., if rating > 3, recommend otherwise not.

To build the model, we took inspiration from famous solution of Netflix Recommendation problem by Simon Funk, and quantified the formula as below

For a given user i and product j, the rating is given by

R = Bi + Cj + U.VT + X.WT

where

Bi = User Bias for user i (scalar)

Cj = Product Bias for product j (scalar)

U = User Vector for user I (vector of size (1 X k))

V = Product Vector for user j (vector of size (1 X k))

X = vector with features other than user id, product id and rating (vector of size (1 X f))

W = weight matrix (vector of size (f X 1))

Other than Mu and X, all the parameters are to be learned using SVD + SGD and/or Machine Learning.

Proposed Tech Stack :

100 – 1000 char

Language: Python

Platform: Jupyter notebook, Google Colaboratory

Frameworks: Scikit Learn, Matplotlib, Numpy, Tensorflow

Notebooks:

Data Exploration.ipynb – Perform EDA on data

Null Values Imputation.ipynb – Remove Null values from data

Train and Test preparation.ipynb – One hot encode the categorical data, and standardize numerical data

SVD.ipynb – Model trained using SVD and SGD

Neural Network.ipynb – First NN, designed to learn only using user\_id, item\_id and rating

Neural Network v2.ipynb – Second NN, designed to learn from user\_id, item\_id, other features and ranking

Market Place/Positioning : (optional)

100 – 250 char

This solution can be positioned as an augmentation to an online shopping portal, where the purchases and ratings of consumers are continuously fed to model so it keeps on learning. The model in turn can be used to recommend new items to consumer, thereby enhancing customer experience.

Effort (Hours) and Cost (INR) of Implementation :

100 – 500 char

Hours

Research: 3-4 hours

EDA (Coding + Analysis): 2-3 hours

Data Cleaning + Vectorization + Verification: 3-4 hours

Model design and training:

1. SVD with SGD : 3-4 hours
2. NN 1 :
   1. Design : 3-4 hours
   2. Hyperparameter tuning and Training : 6-7 hours
3. NN 2 :
   1. Design : 1-2 hours
   2. Hyperparameter tuning and Training : 18-20 hours

PPT