# E-Commerce Products Recommender

**Amr Sherif**
**amrshrif@stanford.edu**

## Introduction & Motivation

E-commerce today plays a vital role in our daily lives, it has redefined the commercial business around the world. Most of the **E-commerce companies** like Amazon, Shein, BestBuy, Zalando, eBay, are using **Recommender systems** for product suggestions to their users and launching promotional campaigns. **35% of Amazon's purchases** are driven by **algorithmic recommendations.**

Amazon is an E-commerce platform that allows users to search and purchase a variety of products (books, electronics, clothing..etc), they can also rate their purchased items on a scale 1 to 5. The **goal is to build a product Recommender** that for any **Amazon customer** its capable to **predict their ratings on products** they didn't purchase and **feature the top 5 products** they will most likely be interested to buy.

## Dataset & Features

The dataset used is a **real-world Product Reviews & Products metadata from Amazon.com**, consisting of a large crawl of product reviews from Amazon. It contains **82.83 million unique reviews**, from around **20 million users and 9 million items** spanning from 1996 to 2014. **Metadata** includes: **reviews and ratings**, item-to-item relationships, timestamps, **helpfulness votes**, **product image** (and CNN features), **price**, **category**, **salesRank.**

| Category | Users | Items | Ratings | Edges |
|---|---|---|---|---|
| Books | 8,201,127 | 1,606,219 | 25,875,237 | 51,276,522 |
| Cell Phones and Accessories | 2,296,534 | 223,680 | 5,929,668 | 4,485,570 |
| Clothing, Shoes and Jewelry | 3,260,278 | 773,465 | 25,361,968 | 16,508,162 |
| Digital Music | 490,058 | 91,236 | 950,621 | 1,615,473 |
| Electronics | 4,248,431 | 305,029 | 11,355,142 | 7,500,100 |
| Grocery and Gourmet Food | 774,095 | 120,774 | 1,997,599 | 4,452,989 |
| Home and Kitchen | 2,541,693 | 282,779 | 6,543,736 | 9,240,125 |
| Movies and TV | 2,114,748 | 150,334 | 6,174,098 | 5,474,976 |
| Musical Instruments | 353,983 | 65,588 | 596,095 | 1,719,204 |
| Office Products | 919,512 | 94,820 | 1,514,235 | 3,257,651 |
| Toys and Games | 1,352,110 | 259,290 | 2,386,102 | 13,921,925 |
| Total | 20,980,320 | 5,933,184 | 143,663,229 | 180,827,502 |

**7 Features** were identified as the **independent variables** for our model **(product name, product description, review body, review summary, purchase verified flag, customer id and product id)** that will be used to predict the user's ratings.
The techniques used for **features transformation** are **One hot encoding & Tf-idf Vectorization** have been applied to the **categorical** features & **text** features respectively.

## Evaluation Metrics

The Recommender Objectives:

- **Predict** a customer's **ratings on products** they didn't buy.
- Feature a ranked list of the **top 5 products** they will most likely be interested to buy.

Evaluation Metrics:
• **RMSE** of the predicted ratings.
• **nDCG**:
As the recommendation strategy is to generate a ranked list, and for a given k products, we compute the average nDCG over users.

$$DCG_u = \sum_{i=1}^{k} 2^{p_{ui}-1} / \log_2(i+1)$$
$$iDCG_u = \sum_{i=1}^{k} 2^{r_{ui}-1} / \log_2(j+1)$$
$$nDCG_u = 1/U \sum_{u=1}^{u} DCG_u / iDCG_u$$

where rui is the actual rating at the actual rank i, and pui the actual rating at the predicted rank i.

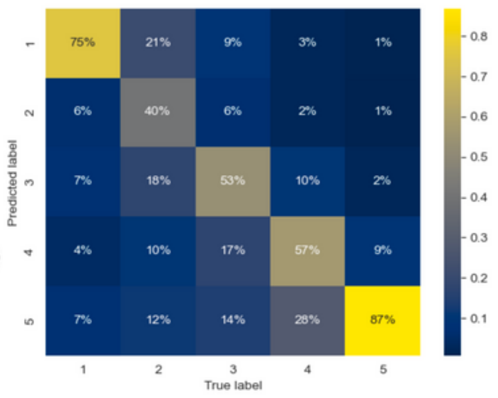
## References

[1] E. Ahmed and M. Moustafa. House price estimation from visual and textual features. NCTA, 2016.
[2] Y. Koren, R. Bell, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. AT and T Labs Research, 2007.
[3] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. IEEE, 2009.
[4] I. MacKenzie. How retailers can keep up with consumers. Mckinsey, 2013.
[5] J. McAuley, C. Targett, J. Shi, and A. van den Hengel. Image-based recommendations on styles and substitutes. SIGIR, 2015.

## Naïve Bayes Multinominal

The **Naïve Bayes Multinominal** model in combination with **Tf-idf** allows us to break these **text descriptions & reviews** into lists of **words** and estimate the **importance of each word** in the text. It combines **Term Frequency** $tf_{bw}$ as number of times a word w appears in a text snippet b divided by total words, and **Inverse Document Frequency** $idf_w = \log(B/df_w)$ as number of text snippets containing w,

$$F_{bw} = tf_{bw} * idf_w.$$




## Collaborative Filtering: Matrix Factorization

Utilizing the concept of **Matrix Factorization,** we can assume the presence of d **latent features**, in which the U x B rating matrix R is represented as the product of two matrices Q and P of sizes B x d and d x U. We are using a biased version of the **SVD algorithm**. therefore the prediction is set as:

$$\hat{r}_{ub} = \mu + b_u + b_b + q_b^T p_u$$

we minimize the following **regularized squared error**:

$$\sum_{r_{ub} \in R} (r_{ub} - \hat{r}_{ub})^2 + \lambda(b_b^2 + b_u^2 + ||q_b||^2 + ||p_u||^2)$$

The update rules:

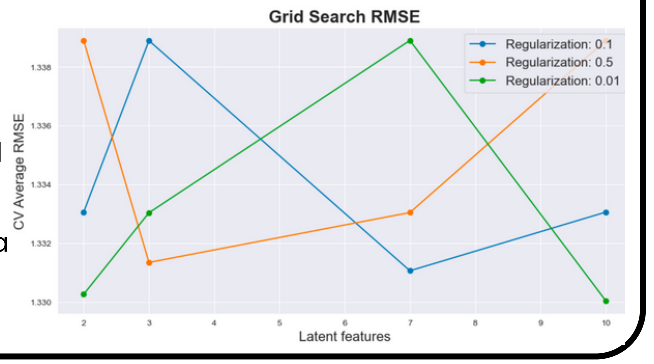
$$b_u \leftarrow b_u + \gamma(e_{ub} - \lambda b_u)$$
$$b_b \leftarrow b_b + \gamma(e_{ub} - \lambda b_b)$$
$$p_u \leftarrow p_u + \gamma(e_{ub} \cdot q_b - \lambda p_u)$$
$$q_b \leftarrow q_b + \gamma(e_{ub} \cdot p_u - \lambda q_b)$$

where $e_{ub} = r_{ub} - \hat{r}_{ub}$.

We used the **grid search technique** over a **cross-validation procedure** with 5 different CV splits to tune the algorithm parameters and found the optimal values for the number of **latent features,** alongside exploring different **regularization** effect and different **initialization technique .**

The best results were obtained using a combination of:

- **10 latent features**
- All **regularization** terms equals **0.01**
- **Learning rate** equals **0.01**
- User and products factors are **randomly initialized** according to a **normal distribution** with **mean** & **standard deviation** equals to **0.01.**




## Discussion & Future Work

Overall we have motivating results we were able to achieve **the lowest RMSE of 0.821** on test set with **Naive Bayes Multinominal** model in combination with **Tf-idf**, **Neural network** proved to be robust with **RMSE of 0.865.**
In future experiments we will focus on: **Optimizing an enhanced version of our matrix factorization model** by focusing on **hyperparameter** tuning, and exploring better **regularization methods** to test performance on a **higher number of latent features.**

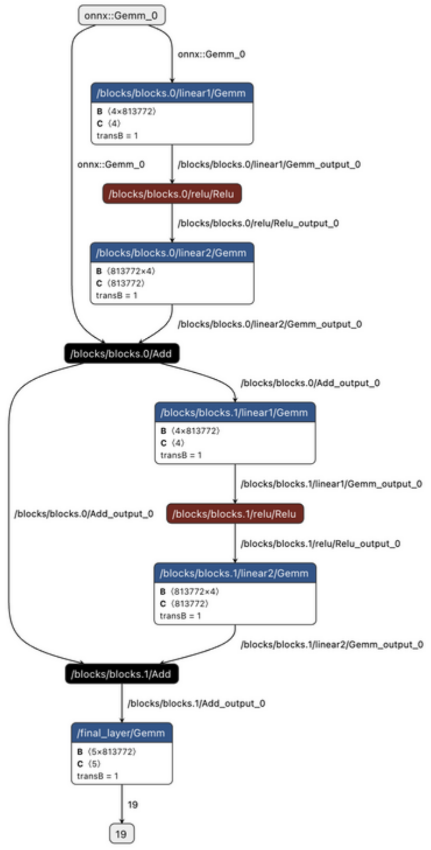
Building a **hybrid model** from our different **collaborative filtering approaches** content/user based to leverage each **model's advantages** into an optimum recommender system.
Leveraging the performance of Neural Networks and incorporate a **multi-modal Network** for **combining the categorical, text, and image data**. By combining different modalities and use separate branches for each modality, followed by concatenating the outputs of each branch and feeding them into a fully connected layer.

## Neural Networks

Using **Neural Networks** to predict ratings on products from the identified features, products metadata, and user's reviews (product name, product description, review body, review summary, purchase verified flag, customer id, product id)

**A Multi Layer Perceptron for the categorical and text data only:**

- Consisting of **2 Residual blocks** each containing **2 fully-connected layers** followed by a **ReLU activation** and the **output layer with softmax activation.**
- Each **layer** consist of **4 neurons** excluding the output layer.
- We have used the **Cross Entropy Loss** as our loss function, along with **Adam optimizer**.
- The dataset had been split into **train/val/test** sets with 0.2 ratio for the val/test sets.
- The needed **transformation** for **the text & categorical features (one-hot encoding & Tf-idf)** has been applied after the split to prevent any data leakage.



## Results

In our results, we focus on **RMSE, nDCG and accuracy.**

| Model | Set | Accuracy | RMSE | nDCG |
|---|---|---|---|---|
| Naive Bayes + Tf-idf | Train | 0.947 | 0.228 | 0.99 |
| Naive Bayes + Tf-idf | Test | 0.733 | 0.821 | 0.99 |
| Matrix Factorization | Train | | 1.133 | - |
| Matrix Factorization | Test | | 1.307 | - |
| MLP | Train | 0.916 | 0.372 | - |
| MLP | Test | 0.734 | 0.865 | 0.99 |

Table 2: Results summary for recommender models

- **Matrix Factorization** didn't perform well to predict users ratings on our dataset, which is **opposite to what was expected**, with **10 latent variables**, we expect to improve its performance in future experiments by doing **more extensive cross validation**, **parameters tuning**, and increase learning iterations which was 20 epochs for this model.
- **Naive Bayes Multinominal** model in combination with **Tf-idf** gave better results than expected with **RMSE of 0.821** on test set, which is better than expected, also **nDCG is very high and close to optimal.**
- **Neural Networks** gave robust results as well on predicting ratings overall with **RMSE of 0.865** and **nDCG is very high** and close to optimal.