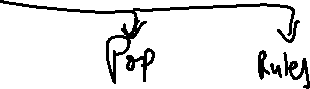
**Recommendation System**

Diagram

Description automatically generated



Latent – Hidden (Wont visible – affects output)

Cold Start Problem -> New User – we don’t know anything; Adding Side Info -> Classification model in the start -> User Attributes ( Location, Age, Gender, Demographics, Credit Card Info, ) -> Popular movies we know they see in our dataset. (Side Information) – classification – first movie recommendation -> User user attribute model.

**Population Avg (Not personalized)**

Avg Rating for movies / business -> Sort recommendation by avg rating independent of how many users rated it.

Global ratings of all movies by all users – avg them – sorting and recommending them to everyone.

**Classification Based**

User Features

Product Features (Director, Plot) -> Classifier -> Like or Not like

Purchase History (What we have bought before)

Run 1 Model per Product (1M Products) -> Scalability

Data availability for Purchase history of every product

**Content Based**

Recommend another movie based on similar description of movie you like, Biz -> Business decriptions

Use Cosine Similarity (Dot Product) [0,1] -> NLP problems

>0.5 similar

Convert Text to Feature -> CountVectorizer / NormalizedCountVectorizer / TF-IDF Vectorizer

Test Doc -> life learning

Doc 1 -> the game of life is a game of everlasting learning

Term Freq for Test Doc

Life learning

1. 1

Normalized TF

0.5 0.5

Doc 2 ->

the game of life is a everlasting learning

1 2 2 1 1 1 1 1

Normalized TF

0.1 0.2 0.2 0.1 0.1 0.1 0.1 0.1

Normalized TF of matched word

Life learning

* 1. 0.1

Cosine Similarity for vectors on same length

Numerator = Dot Product = 0.5\*0.1 + 0.5\*0.1 = 0.1

Denominator = sqrt(0.5^2 + 0.5^2) + sqrt(0.1^2 + 0.1^2) = 0.1

Cosine Similarity = 1

Do the same for another document

Doc – the unexamined life is not worth living

Cosine similarity = 0.707

No Variety if like Romantic Movies – always recommend Romantic Movies.

A picture containing text, screenshot

Description automatically generated

Show1 - [this movie is based on business. Investors pitch to shark Sony money]

Show2 - [this movies is based on investment. Investor visits a business and pitches. Sony money]

Show3 – [this is a thriller where a man kills 20 people and policy is searching. Sony thriller]

Cosine Similarity / Jaccard Similarity

Cosine(1,2) = 0.8

Cosine(1,3) = 0.2

**Nearest Neighbor -**

**User-User Recommendation System**

Diagram

Description automatically generated

UserA 1-4 2-3.5 3-5 4-5

UserB 3-5 4-4.5 5-4 6-5

Cosine Similarity close to 1 for overlapped movies

To expand, Use instead of 1 closest person, use K Neighbors. -> Not watched movies – get avg rating and recommend by sorting order

-🡪 Personalized Recommendation

Issue – Cold Start Problem

Grey Shield Problem -> Hybrid approach, Content + Corraborative

A screenshot of a computer

Description automatically generated with low confidence

A movie recommend to C,

Mean User Rating -> Inherent Bias

**Item-Item Recommendation System**

Recommend items to user that are similar to items user has bought

Application

Description automatically generated with low confidence

Apply Granular level

Diagram

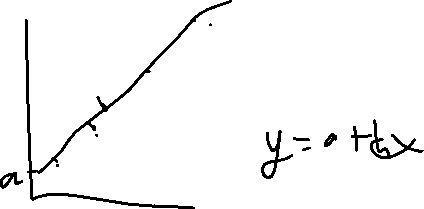
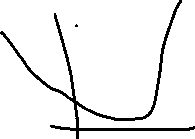
Description automatically generated

Children book – Alice in Wonderland (5\*) -> Content -> all drama only

Item -> Books that are 5\* rated and kids books -> item item in kids section only.

Weighting factors -> Viral Coeff.

**SVD Matrix Factorization**

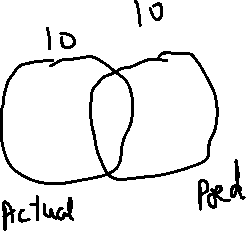


**Evaluation Metrics**

RMSE = Sq rt(Sum( Actual – Pred ^ 2))

Classification Problem:

Top10 predictions – Compare actual preference.



Precision@10 = 3/10 = 0.3

Recall@10 = 3/20 = 0.15

**A/B Test to evaluate Recommendation System on Sample Data before deploying to Production**

Measure values of success metrics

Test

Control

Find Difference between Test and Control

Hypothesis Test

Diff = 0 both systems are same

Diff > - then one system is better than other

**Approach to build Recommendation System**

1. Get Sales / Ratings Data from current Production Data Sources
2. Perform Data Preprocessing to convert data into the form –

**User Product Rating (1-5) / Like (0-1)**

1. Finalize Evaluation criteria such as RMSE for Ratings or Precision@K / Recall@K
2. Run multiple models such as Population Average, Content, Collaborative Filter and evaluate the results on a validation dataset to find the best model.
3. After selecting the right model, use Hyperparameter tuning to figure out the hyperparameters for the recommendation model using Cross Validation
4. You can also use Hybrid model approach by combining results from Top 2 Models and averaging the scores to get an even better model. Make sure latency (time to get results) is low on such model
5. Deploy the model in Production for a Sample of population and measure how your success / guardrail metrics change on this Sample compared to another Sample with No Recommendations. Run A/B Test to check for Statistical difference and if you see good results, deploy the model in your Production System.
6. Decide on an approach to retrain the model based on frequency with which users watch content and the frequency with which new content / product shows up on your feed.