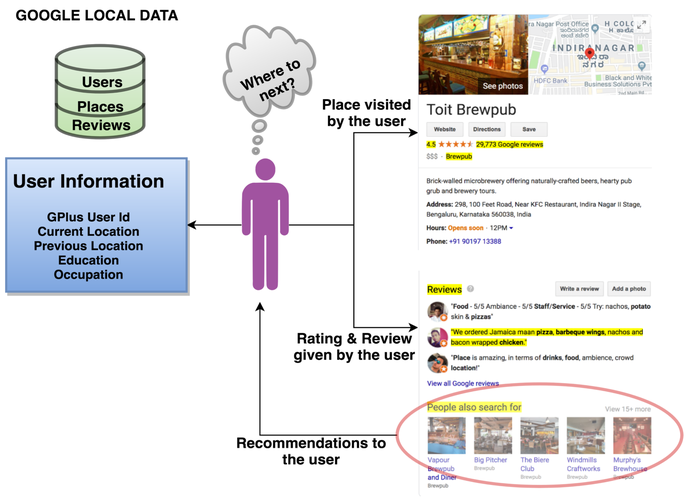
WHERE TO NEXT?: A Recommendation System

V J Kamala, Madhavi Srinivasan, Yash Parikh

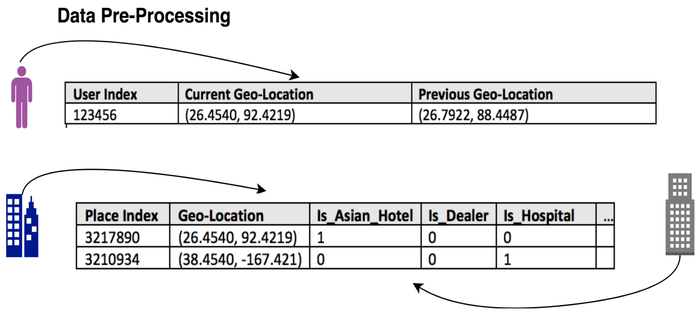
INTRODUCTION:

- Traditional recommendation systems use collaborative filtering and content based methods but don't incorporate time information.
- We aim to create a time-aware recommendation system that also addresses cold start problem effectively.
- We have used the Google Local dataset to recommend places that a user should visit next.

DATA



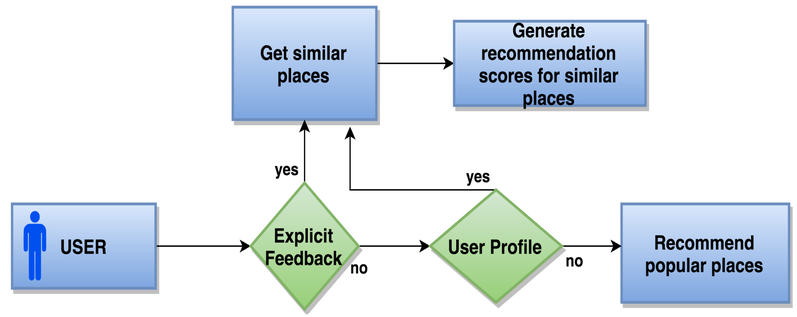
DATA PRE-PROCESSING



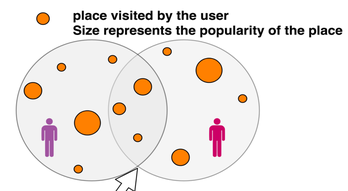
User Index	Current Geo-Location	Previous Geo-Location
123456	(26.4540, 92.4219)	(26.7922, 88.4487)

Place Index	Geo-Location	Is_Asian_Hotel	Is_Dealer	Is_Hospital	...
3217890	(26.4540, 92.4219)	1	0	0	
3210934	(38.4540, -167.421)	0	0	1	

CONTENT BASED FILTERING



COLLABORATIVE FILTERING



$$S_{uv} = \frac{\sum_{s \in S_{uv}} w_s.(r_{us} - \overline{r_u}).(r_{vs} - \overline{r_v})}{\sqrt{\sum_{s \in S_{uv}} w_s.(r_{us} - \overline{r_u})^2}\sqrt{\sum_{s \in S_{uv}} w_s.(r_{us} - \overline{r_u})^2}} \times \frac{min\{I_u \cap I_v\}, \beta}{\beta}$$

discount factor

without time

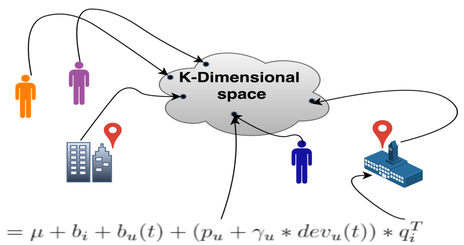
$$\hat{r_{uj}} = \overline{r_u} + \frac{\sum_{K_neighbors} S_{uv} * (r_{vj} - \overline{r_v})}{\sum_{K_neighbors} S_{uv}}$$

with time

$$\hat{r_{uj}} = \overline{r_u} + \frac{\sum_{K_neighbors} S_{uv} * (r_{vj} - \overline{r_v}) w_{vj}(t)}{\sum_{K_neighbors} S_{uv} * w_{vj}(t)}$$

$$w_{vj}(t) = \exp^{-\beta * (t - t_{vj})}$$ Time factor

LATENT FACTOR MODEL



$$\hat{r_{ui}} = \mu + b_i + b_u(t) + (p_u + \gamma_u * dev_u(t)) * q_i^T$$

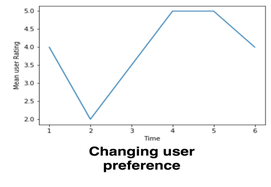
global mean

Item bias

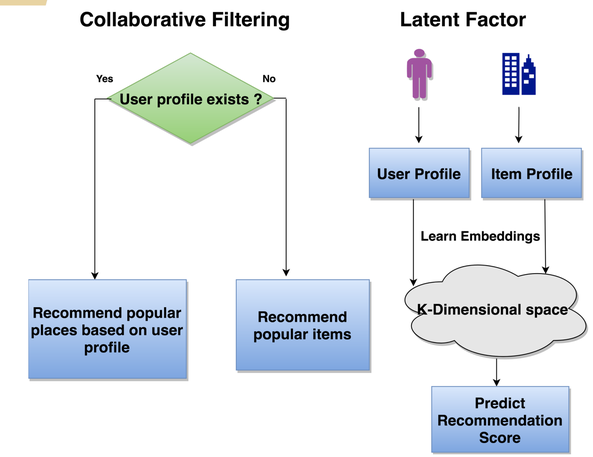
$$b_u(t) = b_u + \alpha_u dev_u(t)$$

User bias

Changing user strictness



COLD START



HYBRID MODEL

β_0 [Latent Factor model (Time aware)] + β_1 [Neighborhood model (Time aware)] + β_2 [Content Based model] = Predicted Rating

β_0 = 0.63 β_1 = 0.36 β_2 = 0.01

RESULTS

Model	RMSE
Basic Latent Factor	1.267
Time aware Latent Factor	1.098
Basic Neighborhood	1.283
Time aware Neighborhood	1.168
Content based filtering	2.361
Hybrid model	**1.065**

FUTURE WORK

- Facilitate a smooth transition of recommendations for cold start users.
- Incorporate sequential recommendation in the hybrid model. Eg: chain of restaurants.
- Get more information about user preferences to enrich the model (based on their search history).