**Section 1:**

|  |  |  |
| --- | --- | --- |
| First Name | Last Name | Email Id |
| Sajal | Jain | sajalj2017@email.iimcal.ac.in |
| Gautam | Kumar | gautamk2017@email.iimcal.ac.in |
| Jaideep | Karkhanis | jaideepk2017@email.iimcal.ac.in |
| Jayanta | Mandi | jayantam2017@email.iimcal.ac.in |

* Submission file name: DSC\_9062427265.R
* Approach Used: MATRIX FACTORIZATION (SVD based recommender system)

**Section 2:**

* Approaches considered:

1. Global top 10 merchants in each category by no. of transactions are found out. For each customer, each category is given weightage as per no. of transactions in that category / total no. of transactions. Top merchants in that category are suggested based on the weight of that category after discarding merchants with whom customer has already transacted
2. Binning carried out - generated lot more insights.

2 important observations :-

* data seems to be time (quarter) independant.
* market share for any merchant is <26%. This means no merchant is “globally” popular. This hints towards customer clustering based on common merchants.

1. But we need to provide some weightage to the first quarter as we need to predict for next 3 months. So effectively we are given data for 4 quarters and we need to predict for the next quarter.
2. Thus a detailed time series analysis doesn’t seems to be appropriate. (¼ th of the total time duration we need to predict). However we should give some importance to transactions in the first quarter of data. There could be cyclicity in terms of annual payments. At the same time we also can not neglect the rest of the quarters as they can also have some cyclicity.
3. Find Similar customer to a particular customer: (clustering/bin based on customers)Take a customer, find customers which spends similar to it. Suggest the merchants of those customers for the customer.
4. Find Similar customers to a particular merchant: Take a merchant, cluster the customers for each individual merchant. And suggest the merchants with which these customer buys

* Approach ultimately chosen:

**SVD Factorization based Collaborative Filtering**

* Steps to approach ultimately chosen, including references to specific lines of your code:

Construct a Sparse Matrix of Customers as rows and Merchants as columns and entries as no. of transactions between a particular customer-merchant pair. Compute the SVD of this Matrix and take a decent rank (say 200) approximation. Check for values which change from 0 in the original matrix to significantly large non-zero values in the reconstructed matrix. The column indices of these values are the potential merchants for a given customer.

* Open source resources and tools used (programming languages) including URLs and references to specific lines of your code:

R has been used for solution.

* Advantages and disadvantages of the approach chosen

Advantages: This approach is similar to the classical SVD based recommender system. Low rank matrix factorization is a very good technique in Collaborative filtering used in recommendation engines.

* Comments on libraries
* Comments on open source resources used : R is widely used for data analytics.
* Special guidance given to algorithm based on training
* Potential improvements to your algorithm