Recommendation System

TVAM, version: 1.0

Introduction

Tvam is a platform which offers various financial, insurance and healthcare/wellness products and solutions to the customers/users. The discussion in the document is limited to the recommendation system which will help Tvam in serving customers and users in much better/personalized ways. It will also aid the platform in achieving continuously improving and lasting relationships with the customers

The goal of the system is following.

- Recommend one or more new/appropriate product/solutions to the customers, largely based on customers' profile, behavior, trends, tendencies, desires, and discovery. At the same time, it should also leverage entities, their relationships, similarities within the context
- It should use Graph, ML, CEP, and continuous stream data analysis techniques for the recommendation purposes
- Should be the unified data source for customer profile, behavior, and other details and allow queries on the same in different ways
- Should be able to assess, learn and improve using past data and outcomes
- It should be an independent scalable service which will cater to the Tvam platform and other applications/services as required
- It should manage the data itself for storage, computations, recommendations, query, and analysis purposes
- It should allow working with the data that is made available to it. And at the same time, it allows future additions/editions to be ingested and processed smoothly (largely autonomous)
- It should be configurable as much as possible; such that different processing logic could be added/edited as and when necessary
- It should be linearly scalable to deliver the expected level of performance
- It can be deployed on cloud or on-prem as desired
- The entire project is quite large in scope and hence we should be able to add and release in staggered manner over a period

The backbone of the recommendation system is in the understanding of the customers. And based on this we must construct set of mechanisms to serve them better. There are some basic unit-level constructs that we define which form the base for rest of the discussion

- Customer profile
- Customer behavior

There are many other entities here, but majority of computations happen on these two items.

Customer profile

The following three broad groups would be created using customer profile.

a) Demographic

Gender
Age - Calculating from DOB
EducationQualification
Monthly_income
Profession –
if Salary – Employment Type, Annual_income, WorkingExp
if Business – Nature of business , Annual turnover, Business turnover
if Agriculture
if Others– College Going or Retired/pensioner or Searching for employment
Noofdependents
MemberChildCount = MaleChildCount +FemaleChildCount
MemberAdultCount = MaleAdultCount + FemaleAdultCount
SeniorCount
MemberCount = MaleChildCount + FemaleChildCount + MaleAdultCount + FemaleAdultCount + SeniorCount
Martial_status
if yes – Started Family
Has_kids
if yes – Kids going to school/college
About to be retired
CreditScore - **
Monthly_balance

b) Geographic

Pincode
City
District
State
lat, lon

c) Psychographic

Hobbies
Interests
Likes/dislikes
Opinions
Goals
types of promotion he/she engages with (does not engage with)

Based on these broad groups, we will create several segments that could be used at run time and/or for other processing

Segments for customer profile groups based on

a) Age

Age < 20	mostly no income – may consume -Payments or investment on small scale – refer
20 < Age < 28	High loan and stock investments
28 < Age < 35	Loan and investments (FD,RD ,MF and stock)
35 < Age < 45	Home Loan and investments (FD,RD and MF)
45 < Age < 58	Home loan or education loan (FD)
58 < Age	

b) Family structure

Single	MemberCount = 0, NoOfdependents = 0
Nuclear	Married + AdultCount = 2, and SeniorCount = 0
Extended	MemberCount > 2, AdultCount > 2 and SeniorCount > 0
SingleParent	Has_Kids – yes , AdultCount = 0 and MemberCount = 1

c) Monthly Income

Monthly income (x)
x < 8000
8000 < x < 16000
16000 < x < 25000
25000 < x < 40000
40000 < x < 60000
60000 < x < 1L
1L < x < 2.5L
2.5L < x

d) Family life cycle

YoungAdults	Age < 20 – College going or Searching for employment and Profession = Others
Independent	Not married + Has_Kids - No + 20 < age < 50,
Marriage	Started Family -Yes + Has_Kids -No + NoOfDepends < 2
Parenting	Has_Kids – yes , Kids going to school, NoOfDepends > 0
AdulParants	Has_kids -yes, Kids going to Collage, NoOfDepends > 0
SeniorYear	Age > 55 , Retired/pensioner, About to be retired

e) Geo-location

Region-	Urban, suburban and rural
Northern	Jammu & Kashmir, Himachal Pradesh, Punjab, Chandigarh, Haryana, Delhi and Rajasthan
Central	Uttarakhand, Uttar Pradesh, Chhattisgarh and Madhya Pradesh
Eastern	Bihar, Sikkim, West Bengal, Jharkhand, Odisha and the Andaman & Nicobar Islands
North-Eastern	Arunachal Pradesh, Nagaland, Manipur, Mizoram, Tripura, Meghalaya and Assam
Western	Gujarat, Daman & Diu, Dadra & Nagar Haveli and Maharashtra
Southern	Andhra Pradesh, Karnataka, Goa, Lakshadweep, Kerala, Tamil Nadu and Puducherry
Further granular groups	Based on District, pin code

f) Interests/hobbies

SportsAndGames
SocialActivities
DomesticActivites
Affilications

Customer Behavior

Following data would be used for behavior computations. We will maintain entity table for the same as well for ready usage of the continuous pre-aggregated data. Most of these data will be part of the Graph itself with different set of relationships between the nodes

Product (APP) usage history

AccountAge	(CurrentDate - CreatedDate)– CustomerDetails [Registration date to Current date]
NoActionPeriod	(CurrentData - LastLogoutDate), RSTAT on this
Session length	Measure the data, do RSTAT Create groups based on these two data points [likes and dislikes]
Session depth	
AvgTimePerVisit	Group by CustRefID (VisitOutDatetime.l – VisitInDatetime.l) , RSTAT on this
SessionsPerWeek (or Day)	Count(CustRefID) groupby CustRefID, RSTAT on this
AvgTimebetweenVisit	Groupby CustRefID (VisitInDateTime.l - VisitOutDateTime.p), RSTAT on this
Survey	Participations, actions, /per survey type
SatisfactionScore	Survey Table
Recency	Last purchase date [curdate – lpd] Recent purchase items Frequency = Count groupby CustRefID [computed at per day/week level] Total Amount Spend = RSTAT on AmountSpend groupby CustRefID, prod category
Frequency	
AmountSpend	
Recency index	# of session with more than n page views in last m weeks(days)/ # total sessions
Click-through-Rate	Number of times clicked to the number of times email or Notification shown, groupby email category, sentiments [groupby emails, notifications with their category if available]
Visited page/product	Records, RSTAT by page, month with daily granularity
Purchase events	Records, RSTAT by product, month with daily granularity
Clickstream data capture	Events as timeseries data into the stream processing engine
ReferralRate	
ConvertionRate	PurchaseCount/TotalVisits

Purchase history

Services_used : (UPI, MPR,Loan,Insurance, EC, Business, investment)	
UPI UsedCount TotalAmount Avg(MonthlyCount) Avg(MonthlyAmount)	Investment MFCount - investedAmount RDCount - investedAmount FDCount – investedAmount Stock – investedAmount Avg(MonthlyinvestAmount)
EC	Business

Booking = 6 monthsCount Amount	UsedfrequencyMonthly only for business users
Insurance InsuranceCount InsuranceTypes insuranceAmount	Loan LoanCount LoanType LoanAmount LoanDuration
MPR Count MPRAmount	BillPayments BillMonthlyAmount

Patterns for customer behaviors for various computations

We would like to compute and keep following ready for consumptions at run time. These will be available in details and a score would be computed for engagement and loyalty.

Engagement

Engagement Pattern	
Active_Users	AvgTimePerVisit (normal to low) – 1200 to 600 sec AvgTimeBtwVisit (low) within 43200 sec to 86400 sec SessionPerWeek (high)- 30 +
SemiActive_Users	AvgTimePerVisit (high to normal) –1200 to 600 sec AvgTimeBtwVisit (normal) 2 to 3 days SessionPerWeek (normal to low) less than 5
Passive_Users	AvgTimePerVisit (high) – high 3600 sec AvgTimeBtwVisit(high) – 3 + days SessionPerWeek (Low) – 1 to 3
Potential_Buyers	AvgTimePerVisit increases and AvgTimeBtwVisit decreases cep

RFM

Based on RFM (From Recency, frequency and Spend scores)	
Best_User	Most recent – most frequent – most Spend
New_customers	Most recent – Low frequency – low spend
High_Spend_New _Users	Most recent – Low frequency – High Spend
Active_Loyal_Users	Recent – high Frequency – normal to low spend
Churned_Users	Not recent – high to normal frequency – normal spend

Survey

Based on Survey [good for collaborative filtering]	
Promoter	Survey – Yes = SatisfationScore above 8
Passives	Survey – Yes = SatisfationScore (5 to 8) or Suvey - No
Detractors	(Survey -Yes) = SatisfationScore (below 5)

Loyalty Index

Click depth index	# of session with more than n page views/ #total sessions
Duration index	# of sessions more than n min / #total num of sessions (or min for all sessions)
Recency index	# of session with more than n page views in last m weeks(days)/ # total sessions
Interaction index	# of session where visitor completed any or tracked activity / # total sessions
Loyalty index	Σ(ci+di+ri+li) , per visitor

Seasonality

Visited page/product	RSTAT by page, month/week
Purchase events	RSTAT by product, month/week
AvgTimePerVisit	RSTAT on this by month/week
SessionsPerWeek	RSTAT on this by month/week

Lifetime value

Total amount spent	RSTAT by page, month/week
Number of products purchased	RSTAT by product, month/week
Number of months as registered user	
Amount of time spent on app	

Nurture index

Repeat visit index	# of repeat visits / #visits
Email perf index	# email interactions /#emails

Structuring of data

We are basically dealing with several entities and some interactions between those entities with different properties. While recommending the products or solutions, we must ensure than we improve the customer's/user's experience. Therefore, it's imperative that we have solid understanding of not just the entities but also the context in which they exist. Hence when it comes to storing the data, we need to ensure that

- We have the context also stored in an efficient manner along with entities [isolated non-related structure is not sufficient, need relations]
- We need relationships or linking of entities but at the same time we need to have scalable and flexible model [Can't use RDMBS, need to break the cold coupling]
- We should be able to add as many links (relations) as required without changing anything which is existing
- We should be able to query the database using these relationships along with various other constraints related to entities and relations properties
- We should be able to pre-compute several properties and attach these entities so that we could query and use them at run time
- The structure should keep evolving as more and more entities and relations arrives
- We should be able to find related entities given a set of entities and set of relationships
- we should be able to find similarities between any give set of entities and use that for recommendation
- We should be able to discover link two isolated entities in an efficient manner
- We should be able to find the complex symmetry or asymmetry between given set of entities
- We should be able to do timeseries data ingestion and processing for continuously update and evolve the structure and keep continuous statistics / aggregation happening for run time efficiency
- We should be able to maintain and continuously update the long-term data for entities such that these could be leveraged at run time
- We should be able to train models on the data existing within the structure such that the models keep updating itself in continual manner and update the existing structure automatically

Therefore, we will have to use combination of following.

A. Graph store for entities and relationships along with various other defined or computed properties along with ML integration for prediction (batch or runtime)
B. Stream processing for ETL, continuous aggregation, running statistics, complex event processing, data-enriching, and long-term entity based pre-aggregated data

Graph Store

The graph store will have entities and they will be connected through relationships.

Entities – immutable id, with flexible/updateable properties

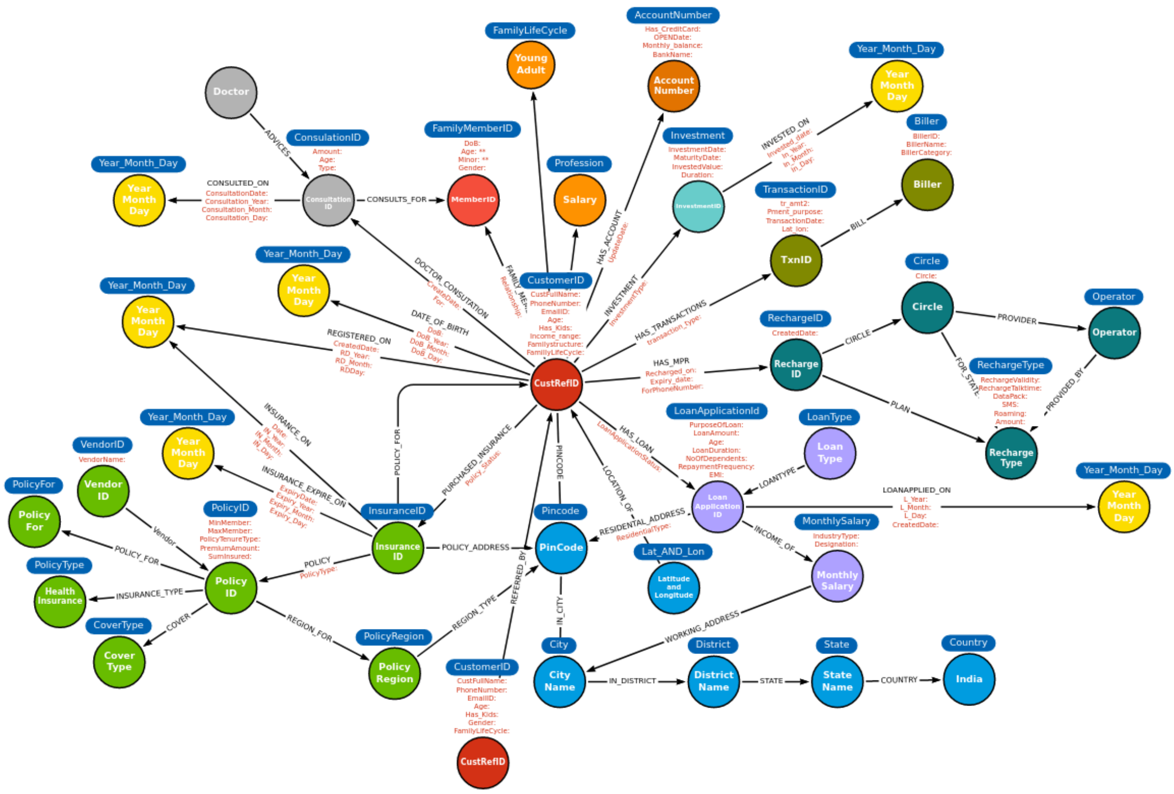
Relations – fluid, flexible, directional, dynamic

There are two basic aspects to the graph structure, raw data graph and computed data graph

a. Basic entities and its relations with several other dissimilar existing entities
b. Entities and relations with other set of other entities with existing relations
c. Entities and relations with several computed entities
d. Entities and relations with other set of entities with computed relations
e. Entities and their relations with similar entities with appropriate strengths
f. Entities and their groupings based on many different sets of dimensions
g. Entities and the clusters

While (a) & (b) are mostly driven by the input data (the raw data). Whereas (c), (d), (e) etc. are data enriched by several mechanisms, like “catr”, “join” etc. by event processing, clustering, similarity, common groupings, community detection through label propagation, path finding, centrality etc. in graph processing along with bringing the whole set of ML algos to the Graph and Event processing as required

Let’s look at the graph structure (subject to change (most likely it will by 20-30%)) which depicts largely (a) & (b) as defined above



ETL, running stats, continuous analysis using “Event processing”

One of the most difficult and heavy tasks for such processing is ETL and it’s widely recognized that sometimes we spent majority of our time in ETL to try to get it right otherwise heavy penalty awaits if even a minor thing goes wrong at the beginning. BangDB avoids this process to great extent by implementing continuous data ingestion mechanism along with processing that could be done to extract and transform what we need at any given time. Here are some of the benefits of using this for the project

- Load data in batch or event manner, as it is, without requiring changing anything in the data
- Transform the data using schema config (“catr”, computed attribute). There are dozens of built-in default functions to leverage for such transformation. Further we can write our own UDFs for custom transformations as required. Going beyond, we can simply attach ML model to add predicted value as part of the event
- We can run CEP to find interesting patterns (in absolute manner) and send the data to another stream, graph as needed. This hugely enriches the data structure
- We can join, refer, and filter events based on queries, conditions, and state to enrich the data structure
- Continuously update the underlying Graph Store as data arrives. Since we have plenty of methods to add/transform the data within event processing framework, the graph receives lot more enriched data along with raw events which makes the structure way more valuable
- Running statistics could be simply used for different attributes and aggregated over different dimensions for as low as second granularity (to min, hour, and any length of time). These can be further rolled up as we need and used for various queries. Note that, doing this on normal database is super costly affair, but it’s extremely fast and efficient due to continuous processing vs batch processing (SQL or MapR)
- Stream maintains even entity table for long term aggregated data along many different dimensions which are readily available for consumption, which otherwise takes hours and days to compute
- Time-series analysis of data is natural and inbuilt by default, and platform supports taking auto actions as configured

Basic set of streams that we will be using are as follows. Although it is represented in json and its fluid in nature, therefore it is set to evolve and change

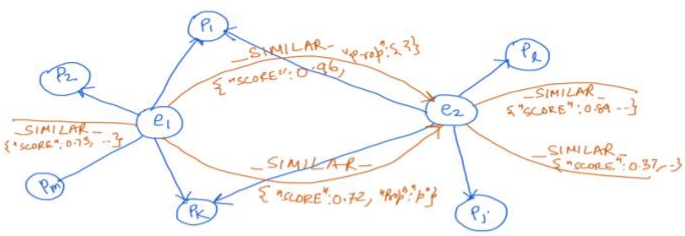
Elements	**Name**	**attributes**
Schema	TvamRecommendation	--
Stream_1	CustomerDetails	8
Stream_2	CustomerAddressDetails	6
Stream_3	CustSurveyFamilyMemberCount	7
Stream_4	CustomerFamilyMember	3
Stream_5	CustomerInsuranceAdditionalDetails	5

Stream_6	CustomerInsurancePolicyDetails	13
Stream_8	InsuranceVendorPolicyMaster	16
Stream_9	TvamLoanApplication	4
Stream_10	TvamLoanApplicant	4
Stream_11	TvamLoanPersonalInformation	5
Stream_12	TvamLoanDetails	4
Stream_13	TvamLoanEmploymentDetails	9
Stream_14	MPRTransaction	8
Stream_15	MPRRechargePlanMasterData	14
Stream_16	UPITransactionDetails	5
Stream_17	ECCustomerDetails	5
Stream_18	ECTvamDoctorConsultation	8
Stream_19	Contacts	5
Stream_20	MRandUPI_entity_stream	5

Concepts and algorithm

1. Entities cluster Analysis within Graph and similarity scores. This is template for similarity based on feature set X

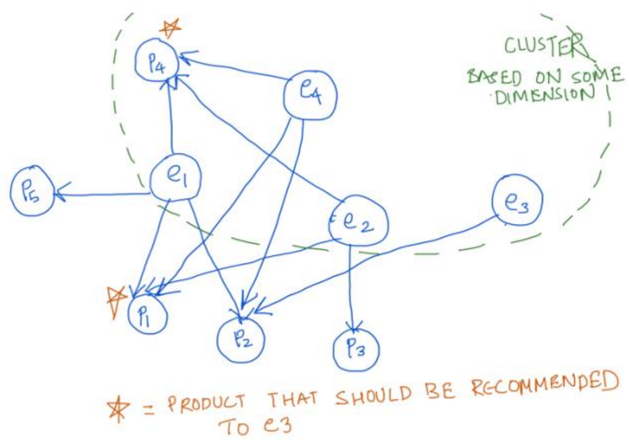
    a. The entities along with relationships and properties in many different combinations will be used to find out the clusters within the data using K-means
    b. Then further similarity scores will be computed based on various assigned centroids and distance from them, frequency, and aggregations of various features along with few configurable inputs
    c. These scores would further be stored in the graph for run time usage for various computations
    d. The entire processes would be continual in nature, and it will self-adjust the scores as more data arrives



Different dimensions [feature set]
1. For insurance product :- PolicyType, CoverType, PremiumAmount, Suminsured, MaxMembers, MinMembers
2. For Inverments: - InvestmentType , riskscore, Subtypes, 1y return, 3y returm, 5y return, Sector, duration
3. Investment Purchase Similarity: - SumAmount(MF), SumAmount(FD) , SumAmount(RD) , SumAmount(Stocks), Avg(MonthlyInvestedAmount)
4. Customer Purchase Similarity: Avg(MonthlyUPICount), Avg(MonthlyUPIAmount), Total(InvestedAmount), Total(LoanAmount), Total(InsuranceAmount) and Avg(MonthlyBillAmount), EC(Avg(Count)Monthly).
5. Customer Similarity based on AccountAge, CreditScore , Monthly_balance and Monthly_income
6. Customer Similarity based on Purchase similarity score, Avg(MonthlySpend) and AccountAge

2. Association rule mining using natural Graph properties

    a. Association rule mining is used to identify the relationships among the set of items/entities within the database. these associations are based on co-occurrence of the data items. Therefore, main purpose of the association rules is to find out the synchronous relationships by analyzing the data and use this as reference during decision making
    b. Collection of items, A = {a1, a2, ...} and group of transactions T = {t1, t2, ...} (Set of items) would be stored in the graph as entities and relations such that the value Conf is computed and stored in the graph for the purpose of finding if X is purchased, then Y can also be purchased with the confidence Conf. The confidence is a measure of number of evidence of the events when X is bought then Y is also bought
    c. The scores/Conf will be stored in the graph itself such that these can be used at run time for decision making. Again, this will also be self-maintaining as we will schedule the computations for re-eval and update
    d. Further we will attach model trained using Apriori algorithm within the graph for run time predictions



1. Set of products/services that are typically bought together
2. MF ,Stocks, FD and RD recommendation based on investment level

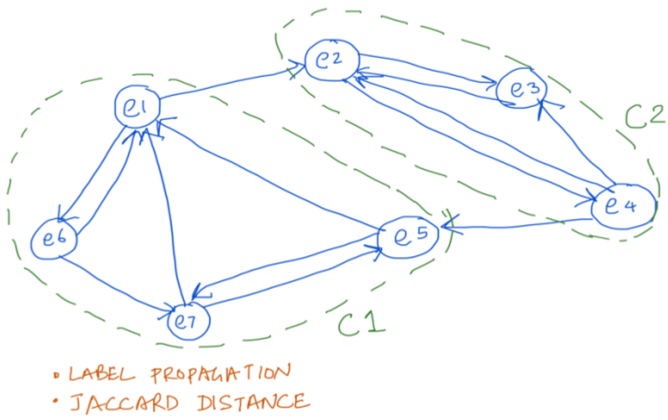
3. Customer segmentation based on cluster analysis

    a. Customer segmentation is the process of grouping customers together based on common characteristics. The natural graph structuring will provide much richer feature set for the training purposes where the entire effort of feature selection, cleaning, correlation, etc. will be avoided. Further models could be trained automatically with defined frequencies and linked within the graph for run time predictions as well
    b. Graph inbuilt clustering mechanisms, groups identifications should be leveraged for this as these are evolving continuously and very fast in nature.

We can also train ML models explicitly using the set of the features already pre-baked in the graph

c. For recommendation engine_1 we are planning to cluster users based on their Language, Gender, Age, City, State, dependency, Marriage status, and education. The result of clustering will be given to the collaborative filtering recommendation system.

d. Mix of Centroid-based K-means Clustering, Centroid-based Mini-Batch K-means Clustering and Distribution-based Gaussian mixture model will be used



finding cluster based on app usage

feature set1 – AccountAge, NoActionPeriod, SessionLength, sessionsPerWeek AvgTimeBetweenVisits, NoPagesVisit

Finding Cluster based on customer spend with respect to age of account

Feature set2 – AccountAge, AmountSpend, PurchaseEvent, ConvertionRate, NoOfProducts

Finding Clusters based on Amount spend on different services.

Feature set3 – Age,Income, TotalAmountInvested, TotalLoanAmount, TotalSpend, Monthly_balance, Count(ActiveInsurance), Total(insurancePermium) and Avg(MonthlyUPISpend)

4. Collaborative filtering for set of features which has fixed and limited set of values to identify similar users – coupled with graph it makes it very effective

   a. Collaborative filtering filters information by using the interactions and data collected by the system from other users. It's based on the idea that people like things similar to other things they like, and things that are liked by other people with similar taste. Graph query will be used for the larger purposes however for certain cases we will also train model and use that for finding the similar profiles

   b. The Collaborative filter technique works by building a matrix for attributes of the user. The matrix is then used to match users with relevant interests and preferences by calculating similarities between user profiles. A group of such users is called a neighbourhood. A user receives recommendations for products that have been brought by the neighbourhood.

c. Without the presence of graph, collaborative based recommenders take time to learn about new users and their relationships with each other or with other entities. Graph provides the data in pre-baked manner for it to be efficient from the start

5. Classification model

   a. For Loan recommendation we will be using a Classification model to predict the probability of a customer to apply for a loan. They will recommend a loan based on the probability value from the model.
   b. For the dataset we know that approx. 1.5 % of the customers apply for a loan. The dataset is small and highly imbalanced. Therefore, for the classification model will use

      1. Logistic Regression - simple and efficient algorithm. Provides probability score for observations.
      2. Naive Bayes - performs better on less training data, very fast
      3. SVM - can work well on small data and very efficient
      4. Decision Tree

Attributes
Gender, Age, monthly_income, Profession, CreditScore, NumberOFCreditCards, Location (Frequency based) , Region, Location segment, MemberCount, RFM group, AccountAge, FamilyLifeCycle, FamilyStructure,
Avg(MonthlySpend), Avg(MontlhySpendInvestments), Monthly_balance.

6. Popularity based / Trend based

   a. It is a type of recommendation system which works on the principle of popularity and or anything which is in trend. These systems check about the products which are in trend or are most popular among the users and directly recommend them.
   b. With graph and entity table in stream, we can provide the list of products with high number of sales, views, engagements groupby geo-loc, age range groups etc. These data could be leveraged directly
   c. Further we will train set of models for predicting the recommendations. To make it a little personalized we will use demographic information as part of the feature set as well

7. Seasonal based

   a. Need to use the aggregated data on the week/month wise purchases from the stream entity table and then store these in the graph with additional relations (and their properties)
   b. Graph query to retrieve the data with filters based on current season/month/week for users who made transactions previously

Queries

Few important queries for finding product for recommendation

a. Which product(s) should be recommended based on types of products already purchased or subscribed or viewed?

Steps:

1. For a given customer, first find the products purchased (may consider time based filtering as well)
2. Then list the set of products which have same group/category as of the purchased product
3. Calculate similarity with selected products with respect to purchased product properties using the similarity scoring technique in the graph
4. Use threshold (configurable) to limit the number of products that could be recommended

Cypher example of insurance:
1. (CustRefID:<refpolicy>)-[POLICY]->(policyid:*)

2. (PolicyID:<policyid>)-[<rel>]->(PolicyID:*), here <rel> = {PolicyType,CoverType,Policy_For and PolicyRegion}

3. Calculating similarity between <refpolicy> and listed policies based on PremiumAmount, SumInsured, PolicyTenureType, MaxMembers.

b. Set of products that are typically bought together or generally people have bought along with other common product

Steps:
1. Listing Customers based on same profile (which customer profile) groups and RFM
2. Filtering Customers based on common set of purchased products during a time duration
3. Listing the items which are commonly purchased within the group but not by this person. We will use multiple queries result-set join property
   a. Two Cypher queries, one for listing the product purchased by the group
   b. Another query for product purchased by the person
   c. Subtract the second set from the first one
4. recommending items

c. Based on other similar people in the group(s), common product that's missing which could be recommended? [profile or behavior or both]

Steps:

1. Listing Customers based on same profile (which customer profile) groups and RFM
2. Filtering Customers based on common set of purchased products
3. Listing the items which are commonly purchased within the group but not by this person. We will use multiple queries result-set join property
   a. Two Cypher queries, one for listing the product purchased by the group
   b. Another query for product purchased by the person
   c. Subtract the second set from the first one

4. recommending items with high count value using some threshold

Query example
1. (CustRefID:<refid>)-[<rel>]->(CustRefID:*), here <rel> = {City, Age_range, Income_range, Profession, FamilyStructure and FamilyLifeCycle, RFM}

2. (CustRefID:*)-[<rel>]->(PolicyID:*) - (CustRefID:<refid>)-[<rel>]-(PolicyID:*), here <rel> = {POLICY,MPR,INVESTMENT and LOAN}

3. count(CustomerID) on items – recommending items with high count value using some threshold

d. Which product is popular for his group or among similar people [profile or behavior or both]?

Steps:

Popular based on location
1. Performing Count operation on products group by location using Entity Stream
2. Recommending product with high values

Popular based on customer profile
1. Performing Count operation on products group by City,Age_range, Income_range, FamilyLifeCycle, FamilyStructure using Entity Stream
2. Recommending product with high values

e. Which seasonal product could be recommended?

Steps
1. Performing count operation based on service using Entity stream and setting time duration as following
- For insurance – type of insurance – purchase count month wise
- For UPI – payment week of the month
- investment – typeOfinvestment – Purchase count month wise
-Loan – Purchase count month wise

f. Which product could be recommended based on personal data?

Steps:
1. Listing customers belonging to same profile (For Same geolocation, Age_range, income_range, FamilyLifeCycle and FamilyStructure)
2. Calculating similarity based on AccountAge, CreditScore and Monthly_balance
3. Count items for Customers with high similarity score

Query example
1. [(CustRefId: {Age_range=”Age< 20”, income_range < 8000})-[IN_CITY]->(City:bangalore)]-[FAMILY_STURCTURE]->(FamilySturcture:Single)

2. Calculate similarity based on AccountAge, CreditScore and Monthly_balance

3. (CustRefID:*)-[<rel>]->(<refLabel>:*) where similarity > 0.85, here <rel> = {POLICY,INVESTMENT, MPR } and <relLabel> are labels with respect to relations

4. Count(custref) based on products

g. Based on recent activities, interests, which product can be recommended?

1. Performing Count operation on product purchased by group by engagement pattern over a set time duration using Event Entity Stream
2. Then we can recommend products with high count value.

h. Based on purchase patterns, amount that he/she spends in a period?

Steps:
1. Listing users based on common group set (example Age_range < 20 , income_range < 8000, FamilyLifeCycle = YoungAdults with FamilyStructure = Single)
2. Calculating similarity based on purchase profile+ Avg(MonthlySpend) + AccountAge
3. Recommending products pruchased by similar users

Query
1. (CustRefID:<refid> {Age_range < 20 , income_range < 8000})-[<rel>]->(CustRefID:*), here <rel> = {FamilyLifeCycle = YoungAdults with FamilyStructure = Single}

2. For the listed Customers we will calculating Similarity based on purchase profile+ Avg(MonthlySpend) + AccountAge

3. Recommending products purchased by users with high similarity score

DATA DESCRIPTION

We have received data for modules – Customer, Insurance, Loan, UPI, Mobile Recharge and Doctor Consultation. Assumption for selecting Machine learning algorithms will be based on given data but we will also keep in mind that there is a possibility that the table structure may change from Atyati's end.

Below we have tried to summarize the data we have received. We have listed the tables and their attributes that we will be using followed by the data summary.

1. Table and attributes for first Recommendation model

1.1 Customer Data:-

Table Name : CustomerDetails – 26922 records		
No	**Attribute Name**	**Definition**
1.	CustRefID	Unique customer key
2.	Gender	Categorical Female/Male
3.	DOB	Date time DD/MM/YY HH:MM:SS
4.	Martial_Status	Categorical – Married/ Un-Married
5.	Profession	Categorical – Salaried/Agricultural/Self-Owned/ Other
6.	Monthly-Income	Double data type
7.	Has_Kids	Categorical No/Yes

Table Name : CustomerAddressDetails – 26922 records		
No	**Attribute Name**	**Definition**
1.	CustRefID	Unique customer key
2.	City	Customer city name -demographic
3.	State	Customer state name - demographic
4.	Country	Customer country - demographic
5.	Pincode	Numerical string
6.	District	Customer address - demographic

Table Name : CustSurveyFamilyMemberCount – 26922 records		
No	**Attribute Name**	**Definition**
1.	CustomerID	String, unique key CustRefID = CustomerID
2.	MaleAdultsCount	Count values
3.	FemaleAdultsCount	Count value

4.	MaleChildCount	Count value
5.	FemaleChildCount	Count value
6.	SeniorCitizenCount	Count value
7.	MemberCount	Count Value

Table Name : CustomerFamilyDetails – 26922 records		
No	**Attribute Name**	**Definition**
1.	CustomerID	String, unique key CustRefID = CustomerID
2.	MemberID	String –unique key
3.	Gender	Categorical

1.2 Insurance Data:-

Table Name : CustomerInsuranceAdditionalDetails – 5382 records		
No	**Attribute Name**	**Definition**
1.	InsuranceID	String, unique key for every insurance applied
2.	CustRefID	Customer ID
3.	Pincode	Pincode –string values
4.	District	Location
5.	State	Location

Table Name : CustomerInsurancePolicyDetails – 5382 records		
No	**Attribute Name**	**Definition**
1.	CustomerID	String, customer ref id
2.	InsuranceID	String, insuranceID is similar to Policy type
3.	PolicyStatus	Defining status of the policy. Categorical ex:- INITIATED,SUCCESS/CANCEL/NULL
4.	PolicyDownload	Categorical – 0/1
5.	Policy Type	Categorical -Health or accidental insurance
6.	CoverType	Categorical – FF/individual, whether for family or self
7.	PolicyID	String -unique, each policy have policy id
8.	PolicyTerm	Int Value, For how many years
9.	SumAssured	Policy, amount to be covered by policy = SumInsured
10.	Productcode	Numerical string, policy id from vendor side
11.	PolicyFor	Categorical – SELF/FAMILY/BOTH
12.	PolicyMaturityDt	Maturity date for the policy

1.3 Insurance product Data:-

Table Name : InsuranceVendorPolicyMaster – 22 records		
No	**Attribute Name**	**Definition**
1.	VendorID	String, Unique id for vendors
2.	InsuranceType	Categorical Health, accidental
3.	VendorCode	String -name – ex - CARE01
4.	CoverType	Categorical – whether insurance for family or self
5.	PolicyID	Unique id given to each policy
6.	SumInsured	Amount to be covered by insurance
7.	BenefitURL	Url for pdf
8.	AmountPerMember	Amount to be pay per person
9.	PremiumAmount	Amount for policy
10.	MaxMembers	Max members for policy
11.	PolicyTenuretype	year or month of policy cover
12.	policyTenureValue	Count of year/month covered
13.	PolicyRegion	URBAN/RURAL
14.	MinMember	Minimum members allowed
15.	ProductCode	Unique code for policy
16.	SumInsuredCode	Code for amount covered by insurance =
17.	PolicyBrochureURL	Url for PDF

1.4 Loan Data:-

Table Name : TvamLoanApplication – Same for all Loan application – 300 records		
No	**Attribute Name**	**Definition**
1.	LoanApplicationId	Unique id for every Loan applicant
2.	LoanTypeId	Categorical –PRSL/BUSN
3.	CustomerId	Same as CustRefID
4.	LoanApplicationStatus	Loan status – categorical -PENDING, QUEUED...

Table Name : TvamLoanApplicant – 300 records		
No	**Attribute Name**	**Definition**
1.	LoanApplicationId	Unique id for every Loan applicant
2.	LoanApplicationName	String Applicant name
3.	Gender	Male/Female/other

4.	Age	Long value

Table Name : TvamLoanPersonalInformation – 300 records		
No	**Attribute Name**	**Definaition**
1.	LoanApplicationId	Unique id for every Loan applicant
2.	ResidentialTypeId	Categorical
3.	City	city name -demographic
4.	Pincode	demographic
5.	NoOfDependents	Int count value

Table Name : TvamLoanDetails – 300 records		
No	**Attribute Name**	**Definition**
1.	LoanApplicationId	Unique id for every Loan applicant
2.	PurposeOfLoan	Categorical
3.	LoanAmount	Int value
4.	LoanDuration	Option give to customers in 1 to 5 but converts to month

Table Name : TvamLoanEmploymentDetails - 300 records		
No	**Attribute Name**	**Definition**
1.	LoanApplicationId	Unique id for every Loan applicant
2.	City	city name -demographic- working place
3.	State	Working place
4.	IndustryType	Categorical
5.	MonthlySalary	Double
6.	Designation	Categorical

1.5 Mobile Prepaid recharge:-

Table Name : MPRTransaction – 20921 records		
No	**Attribute Name**	**Definition**
1	CustomerID	String Unique Id
2	CircleID	String
3	OperationID	String
4	Amount	Double

Table Name : MPR Circle – 23 records		
No	**Attribute Name**	**Definition**
1	CircleID	String Unique Id
2	CircleName	String
3	CreatedDateTime	String Date Time
4	UpdatedDateTime	String Date Time

Table Name : MPROperator – 5 records		
No	**Attribute Name**	**Definition**
1.	OperatorID	String Unique Id
2.	OperatorName	String
3.	CreatedDateTime	Date-time

Table Name : Contacts – 439655 records		
No	**Attribute Name**	**Definition**
1.	CustRefId	String Unique Id
2.	ContactName	String
3.	ContactMobileNo	string
4.	IsTvsmUPIRegistered	Categorical – 0/1
4.	IsTvamRegistered	Categorical – 0/1

Table Name : MPRRechargePlanMasterData – 56 records		
No	**Attribute Name**	**Definition**
1.	RechargeId	String ID
2.	Operator	Categorical
3.	Circle	States – Categorical
4.	Amount	Double value
5.	RechargeTalktime	Double
6.	RechargeValidity	String – with N.A
7.	RechargeShortDesc	Sting
8.	RechargeLongDesc	String
9.	RechargeType	Categorical
10.	UpdatedAt	Date-Time
11.	Version	Categorical
12.	Category	Categorical

13.	Data	String -Categorical
14.	CreatedDateTime	Date -Time

1.6 UPI Data:-

Table Name : UPI Transactiondetails – 215826 records		
No	**Attribute Name**	**Definition**
1	TransactionId	String Unique Id
2	TvamCustomerId	String
3	Amount	Double
4	TransactionType	Categorical
5	PaymentPurpose	Categorical

1.7 EC Doctor Consultation:-

Table Name : ECCustomerDetaits – 3089 records		
No	**Attribute Name**	**Definition**
1	TvamCustid	String Unique Id
2	FamilyMemberid	String
3	CustomerGuid	String
4	Gender	Categorical
5	CreatedDate	Date-Time YYYY-MM-DDTHH:MM:SS.

Table Name : ECTvamDoctorConsultation – 3089 records		
No	**Attribute Name**	**Definition**
1	TvamCustID	String Unique Id
2	Type	Categorical
3	UserDob	Have year, month, date
4	UserGender	Categorical
5	USERAge	Long values
6	ConsultationDate	Date -time
7	FamilyMamberId	String Unique Id
8	Createdate	Date-time

2. Data Summary

2.1 Dataset information:

Customer Data		Number of Variables	21
		Number of records	26922
		Total Missing	0%
Insurance Data	12.61% of the customers have taken insurance. 3395 unique customers	Number of Variables	21
		Number of records	5382
		Total Missing	2.1%
Loan Data	1.11% of the customers have taken loan	Number of Variables	19
		Number of records	299
		Total Missing	0%
MR	14486 unique users	Number of Variables	4
		Number of records	20921
		Total Missing	0%
UPI	19500 unique users	Number of Variables	5
		Number of records	215826
		Total Missing	10.0%
EC Customers	11.27% of the customers have taken doctor consultation. 3035 unique users	Number of Variables	8
		Number of records	3089
		Total Missing	20.0%

2.2 Univariate Statistic Summary:

2.2.1. Gender: - Categorical with Distinct Count = 2. Comparing Gender distribution for customers, insurance, loan and EC Customer dataset.

For Customer Dataset

Value	Count	Frequency (%)
Male	18857	70.0%
Female	8065	30.0%

For Insurance Dataset

Value	Count	Frequency (%)
Male	3849	71.5%
Female	1533	28.5%

For Loan Dataset

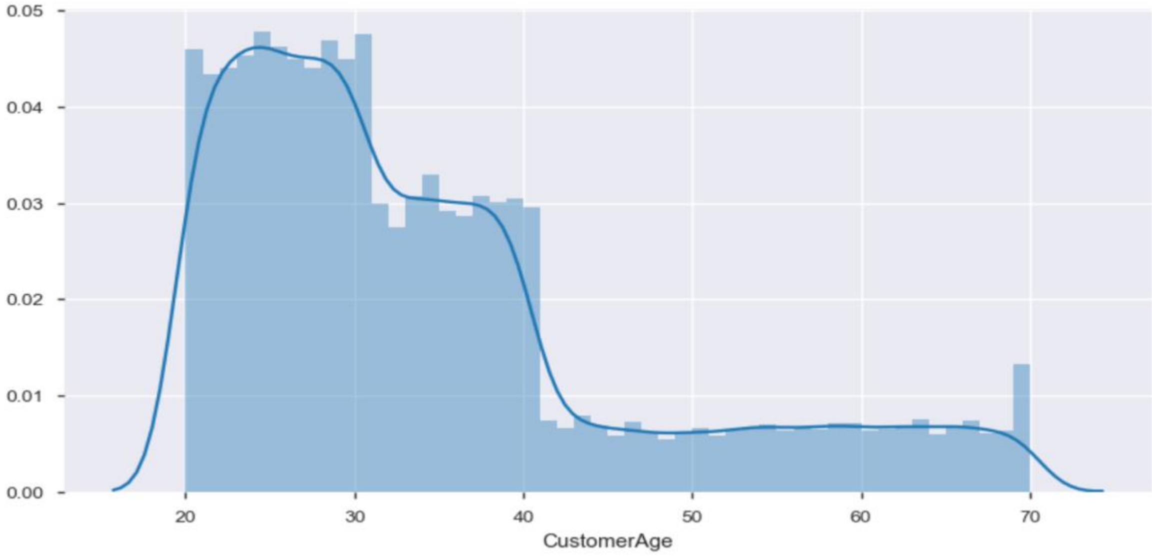
Value	Count	Frequency (%)
Male	208	69.6%
Female	91	30.4%

For EC Customers

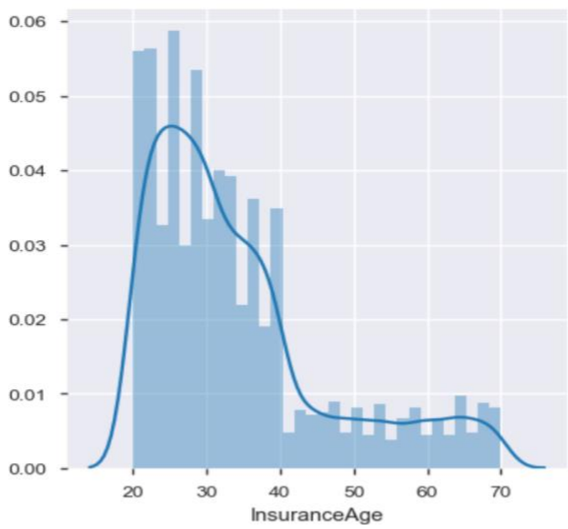
Value	Count	Frequency (%)
Male	1987	64.3%
Female	1102	35.7%

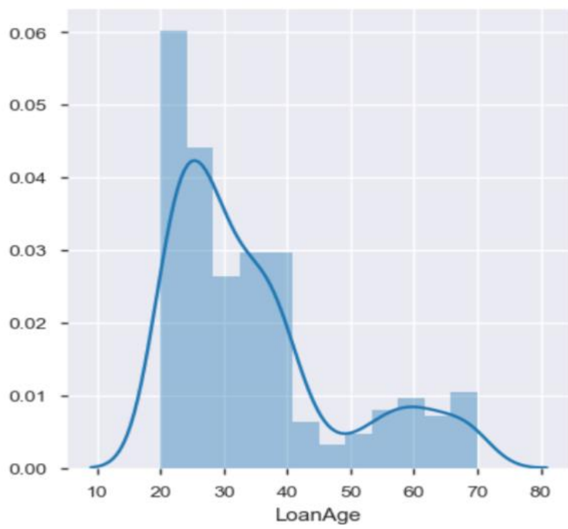
2.2.2. Age – Calculation age from DOB – Comparing Age distribution for Customers, insurance data and Loan data

Customers Dataset: - Mean – 34.248 with minimum = 20 and maximum = 70

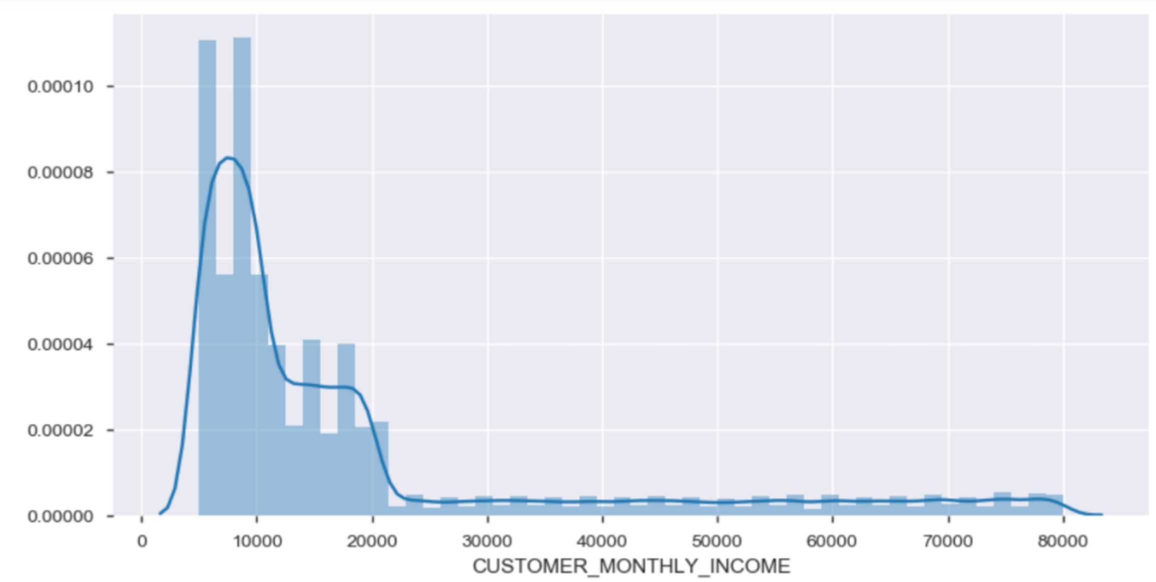


Age distribution for Insurance and Loan Dataset

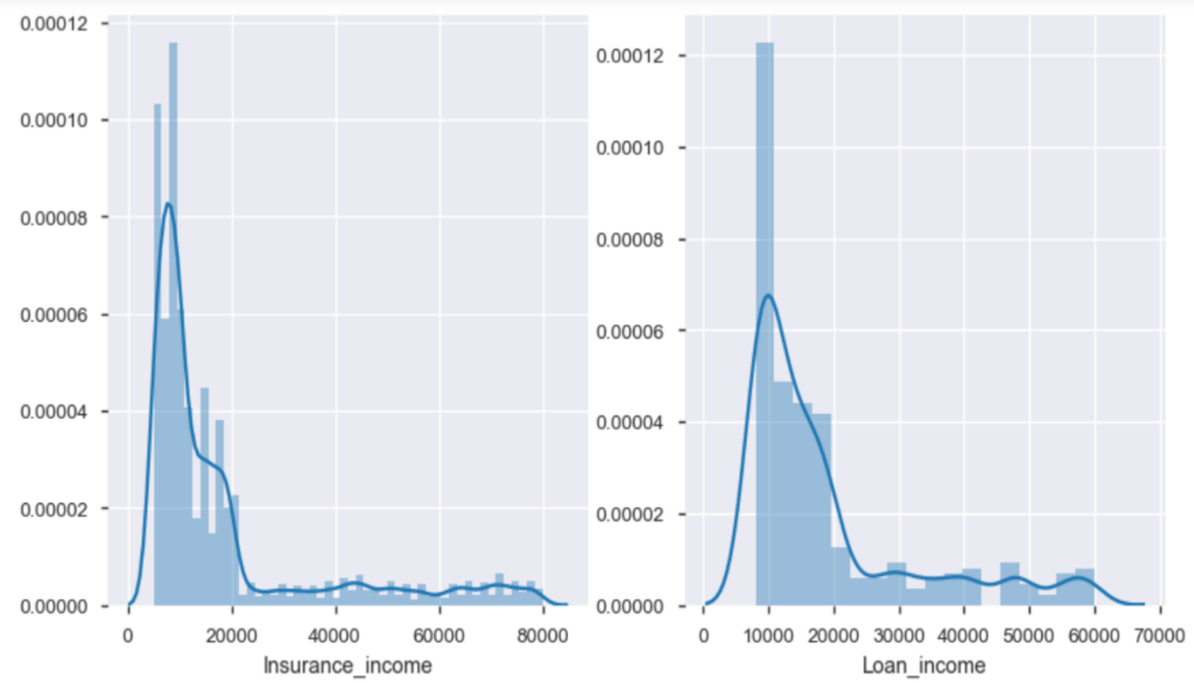




2.2.3. Monthly Income: - Mean – 18,553 with minimum = 5000 and maximum = 80000



Income Distribution for Insurance and Loan Customers



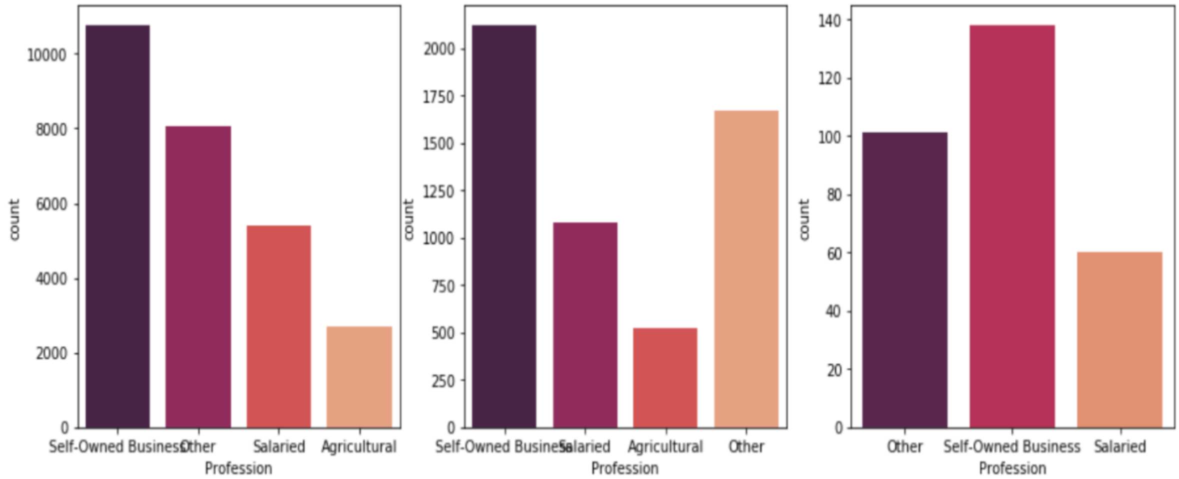
2.2.4. Marital Status – Categorical variable with distinct count 3

Value	Count	Frequency (%)
Married	20773	77.2%
Un-Married	6020	22.4%
Divorced	128	0.5%
(Missing)	1	0.0%

2.2.5. Profession - Categorical variable with distinct count 4

Value	Count	Frequency (%)
Self-Owned Business	10768	40.0%
Other	8076	30.0%
Salaried	5384	20.0%
Agricultural	2692	10.0%
(Missing)	2	0.0%

Profession – For customer data, for insurance data and for Loan data



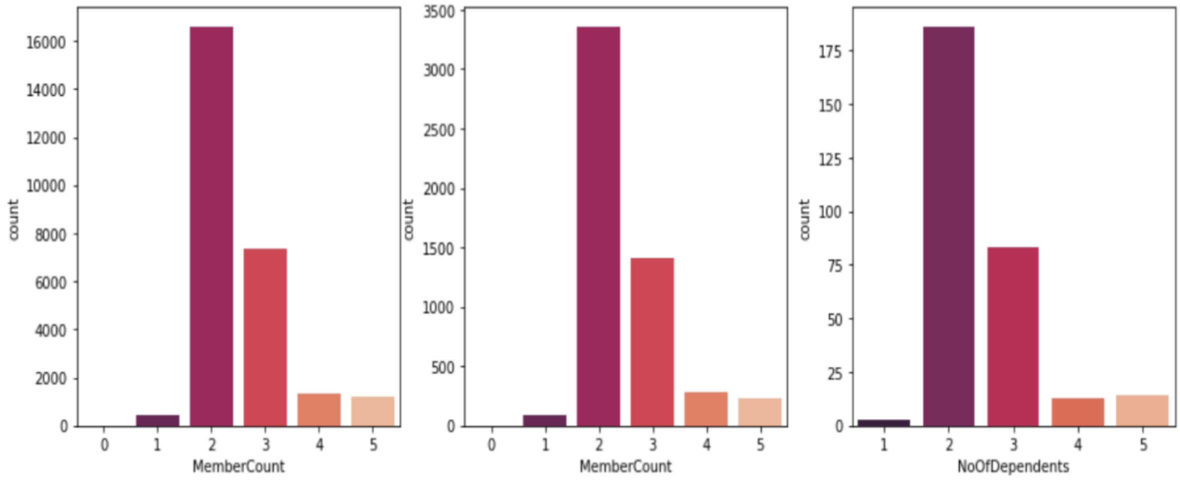
2.2.6. Has_Kids – Categorical variable with distinct count 2

Value	Count	Frequency (%)
No	21216	78.8%
Yes	5706	21.2%

2.2.7. MemberCount – Numeric variable with minimum 0 and Maximum 5

Value	Count	Frequency (%)
2	16591	61.6%
3	7346	27.3%
4	1333	5.0%
5	1203	4.5%
1	440	1.6%
0	9	0.0%

First – For Customer data, Second – For Insurance data and Third – For Loan data



2.2.8. State- Categorical variable with distinct count 26. Below fig show count values customer dataset.

Value	Count	Frequency (%)
West Bengal	9640	35.8%
Bihar	3129	11.6%
Andhra Pradesh	2416	9.0%
Assam	1972	7.3%
Uttar Pradesh	1935	7.2%
Odisha	1407	5.2%
Maharashtra	1138	4.2%
Karnataka	932	3.5%
Tamil Nadu	922	3.4%
Telangana	755	2.8%
Other values (16)	2676	9.9%

2.2.9. Country – Has constant value India

2.2.10. City – Categorical variable with distinct count 5474 – has high cardinality

2.2.11. District – Categorical variable with distinct count 584 – has high cardinality

2.2.12. CoverType – Categorical variable with distinct count 2

Value	Count	Frequency (%)
Individual	2748	51.1%
FF	2634	48.9%

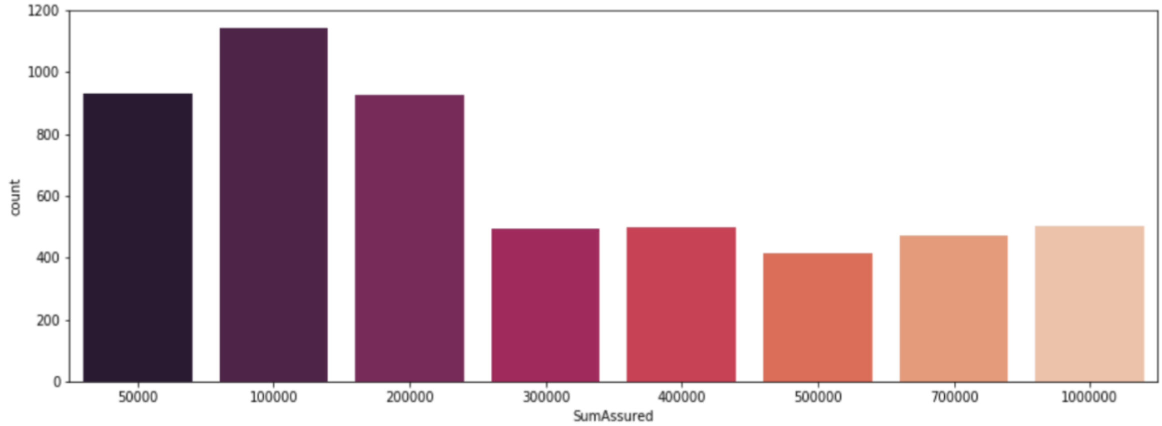
2.2.13. PolicyType – Categorical variable with distinct count 2.

Value	Count	Frequency (%)
HealthInsurance	5171	96.1%
AccidentalInsurance	211	3.9%

2.2.14. PolicyFor – Categorical variable with distinct count 3 and 12.9% missing values (696 missing values)

Value	Count	Frequency (%)
FAMILY	2054	38.2%
BOTH	1965	36.5%
SELF	667	12.4%
(Missing)	696	12.9%

2.2.15. SumAssured –



2.2.16. Expiry Date: - Extracting year

Value	Count	Frequency (%)
2023	2704	50.2%
2022	2678	49.8%

2.2.17. LoanType: - Categorical variable with distinct count 2 (PRSL and BUSN)

Value	Count	Frequency (%)
PRSL	265	88.6%
BUSN	34	11.4%

2.2.18. LoanApplicationStatus: - Categorical variable with "SUCCESS" constant value.

2.2.19. Designation: - Categorical variable with 43 distinct count and Unique % = 14.4%

Value	Count	Frequency (%)
EXECUTIVE DIRECTOR	17	5.7%
ENGINEER	15	5.0%
ACCOUNTANT	12	4.0%
OTHERS	12	4.0%
MANAGER	11	3.7%
SENIOR MANAGER	9	3.0%
AREA SALES MANAGER	8	2.7%
ASSISTANT MANAGER	8	2.7%
DATA ENRTY OPERATOR	8	2.7%
PARTNER	8	2.7%
Other values (33)	191	63.9%

2.2.20. Circle Names: - Categorical variable with distinct count 23

Value	Count	Frequency (%)
Maharashtra	980	4.7%
Haryana	960	4.6%
Himachal Pradesh	952	4.6%
Andhra Pradesh	950	4.5%
North East	949	4.5%
West Bengal	945	4.5%
Jammu Kashmir	922	4.4%
Bihar Jharkhand	916	4.4%
Karnataka	916	4.4%
Rajasthan	911	4.4%
Other values (13)	11520	55.1%

2.2.21. PurposeOfLoan: - Categorical variable with distinct count 49 and unique 16.4%

2.2.22. ResidentialType: - Categorical variable with distinct count 17 and unique 5.7%

2.2.23. IndustryType: - Categorical variable with distinct count 36 and unique 12%

2.2.24. For MR – dataset Operators Categorical variable with distinct count 5

Value	Count	Frequency (%)
Jio	6283	30.0%
Airtel	6237	29.8%
Vi	4299	20.5%
MTNL	2079	9.9%
BSNL	2023	9.7%

2.2.25. Payment purpose: - UPI dataset

Value	Count	Frequency (%)
unknown	89109	41.3%
food	10595	4.9%
others	5300	2.5%
bill	1100	0.5%
movies	1095	0.5%
gift	1044	0.5%
(Missing)	107583	49.8%

2.2.26. Transaction type: - Categorical variable with distinct count 2

Value	Count	Frequency (%)
payer	108243	50.2%
receiver	107583	49.8%

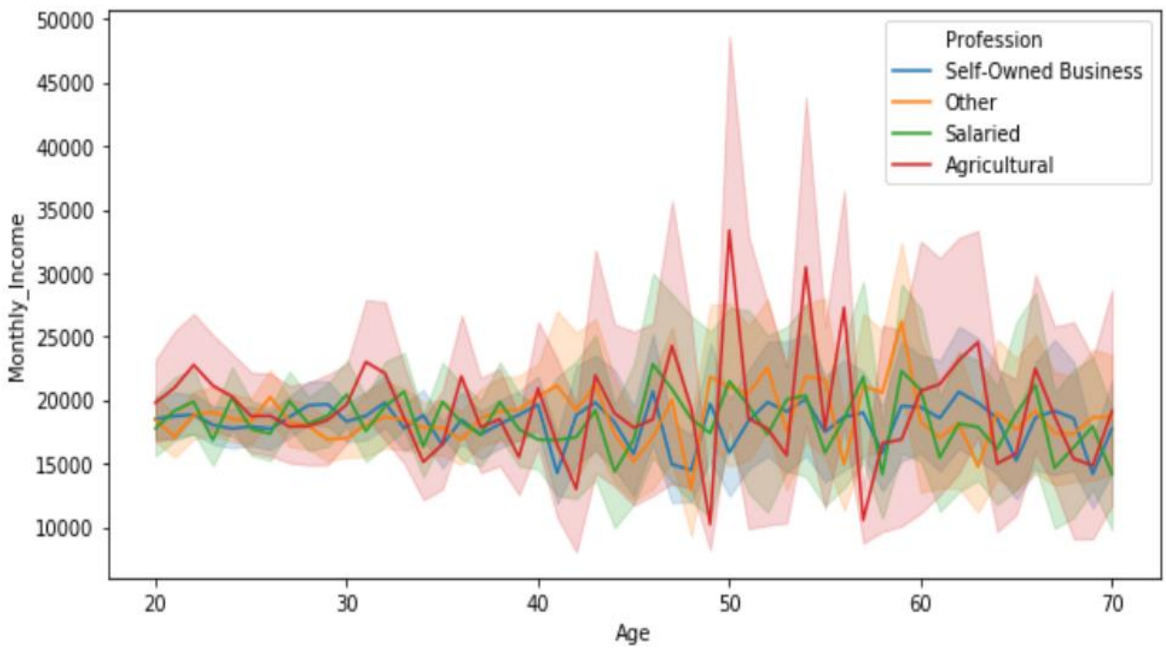
2.2.27 Policy Status: Categorical variable

Value	Count	Frequency (%)
CANCELLED	1396	25.9%
success	1396	25.9%
INITIATED	1334	24.8%
(Missing)	1256	23.3%

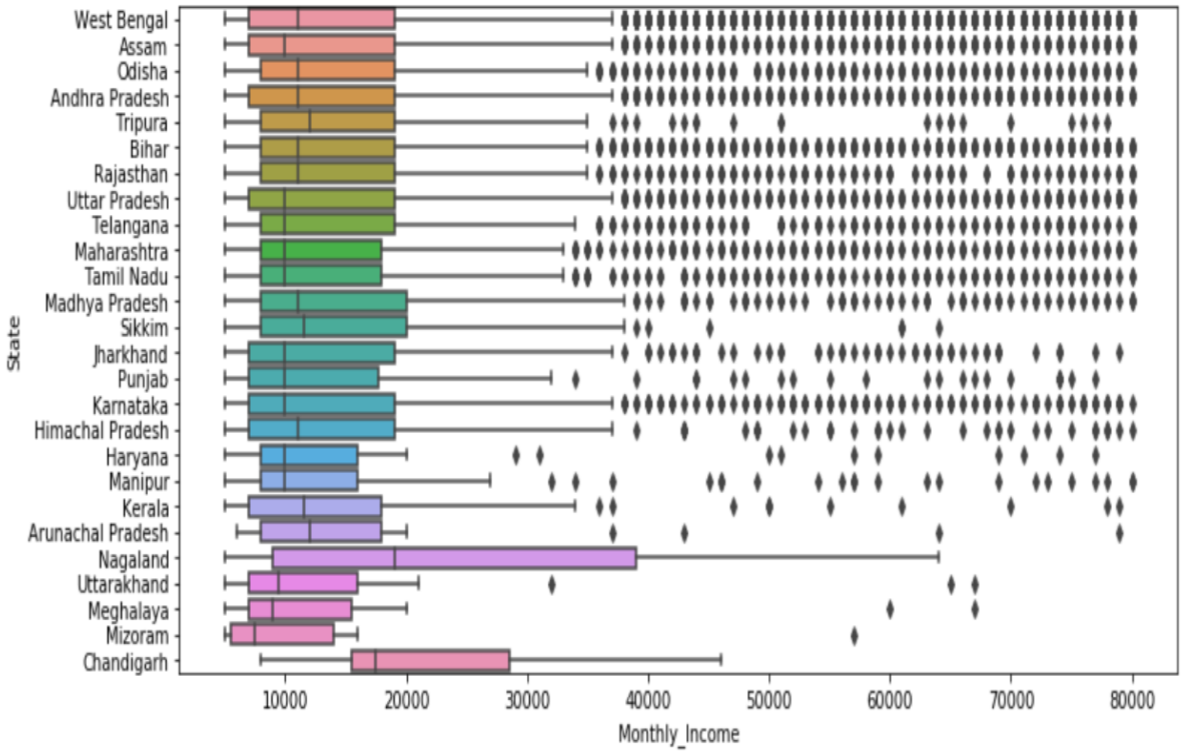
2.2.28. For dataset MPR Plans: we have data from circle – Karnataka

2.3 Bivariate Statistic Summary:

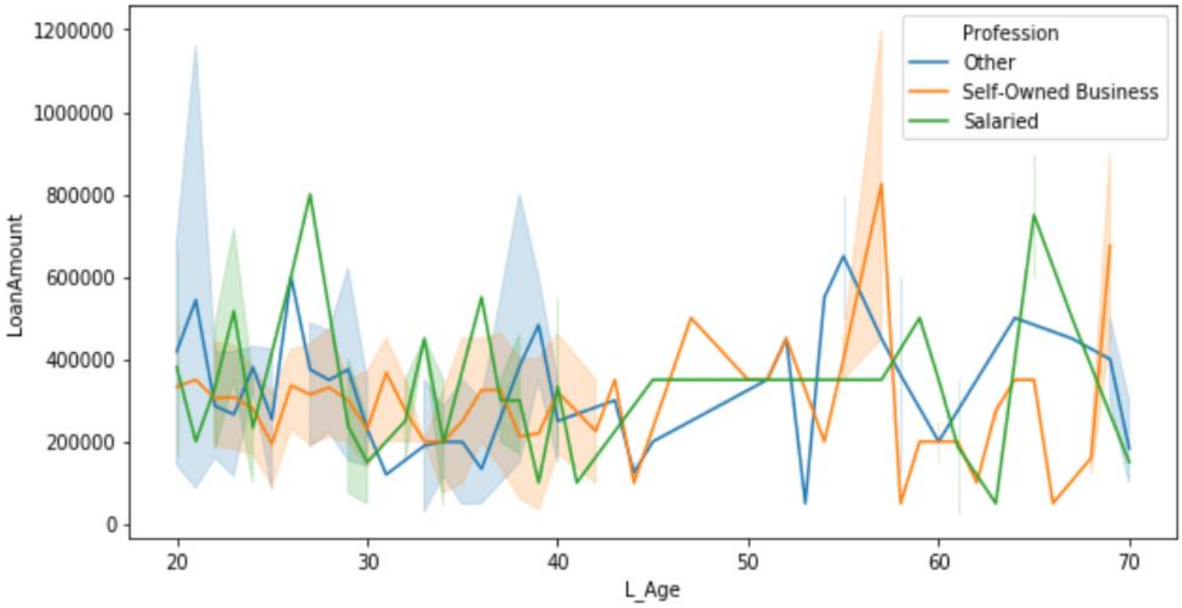
2.3.1. Profession VS Age VS Income



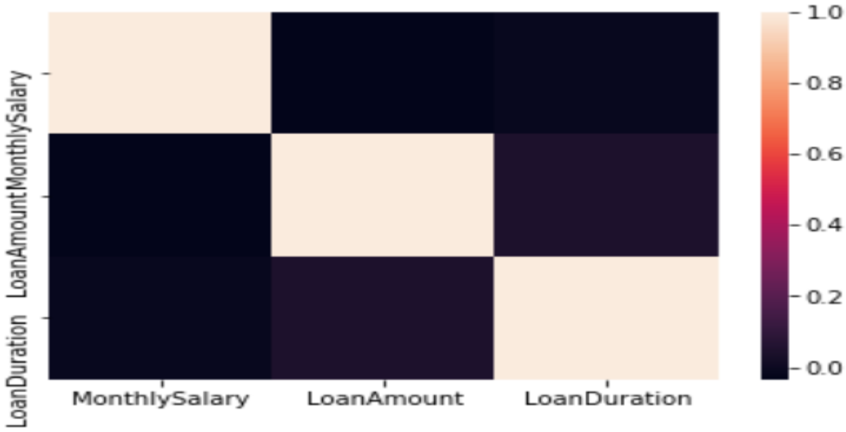
2.3.2. State Vs Income:-



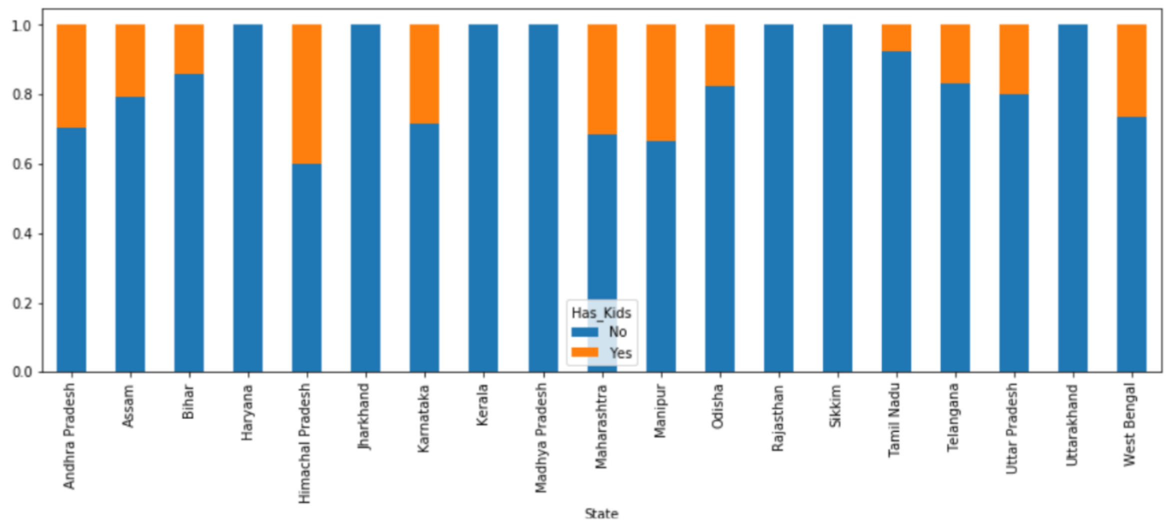
2.3.3. Age Vs LoanAmount Vs Profession



2.3.4. Monthly income Vs Loan Amount Vs LoanDuration – heat map – checking correlation



2.3.5. For Loan Data – State Vs Has Kids



2.3.6 Frequency distribution for 'Number of features a customer is using'

Value	Count	Frequency (%)
2	10883	40.4%
1	10813	40.2%
0	2578	9.6%
3	2461	9.1%
4	182	0.7%
5	5	0.0%

Some insights: - With total customer 26922,

1. 2578 users are not using any feature
2. 5 users are using all features

3. 53% of UPI users are MR users (72% of MR users have UPI)

4. Out of 300 loan users, 224 uses UPI

5. Out of 300 loan users, 43 users also have insurances

6. 373 insurance users also uses EC feature