Predictive Model for Insurance Company

Mentored by: Mr. Gunnvant Saini
Presented by: Group 4, IPBA Batch 2
Presenters –

- Shubhangi Bansal
- Gargi Baser
- Pratik Khadse
- Neela Madhav Suram
- Adeep Bhojne

Overview

- Business Problem
- Data Wrangling(Munging)
- Exploratory Data Analysis
- Model Building
- Market Basket Analysis
- Conclusion

Business Problem

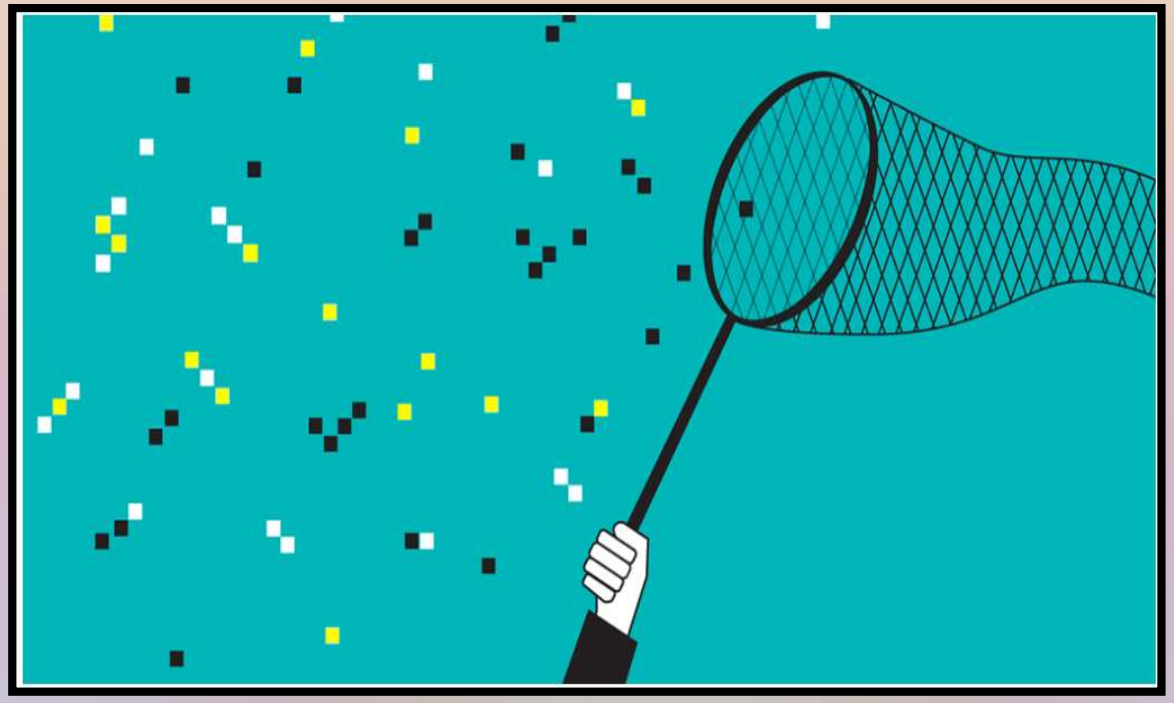
- Client is a leading Insurance Company having long term protection & savings solution plans as their products.
- Cost of acquiring new customer > expanding existing customer
- Client would like to cross-sell their products to their existing customer base to:
  - Gain more revenue
  - Maximize value of customer portfolio

Develop a predictive model:
To identify customers who will
Buy additional policies

Solution

Identify the product/s:
What type of policies are those
Customers most likely to purchase

Data Wrangling (Munging)



Step 1

Performed Variable Identification to identify significance of each variable

Step 4

Perform data cleaning and treatment

Step 2

Compared the two datasets (Merged & CustSegList)

Step 3

Create raw data from data chosen in step 2 to identify Target Variable and perform further analysis

Step 1: Variable Identification

Identity Variables	Policy related Variables
policy_owner_number	policy_number
Own_gender	premium
LA_gender	afyp
Own_Education	sum_assured
Own_Edu	RCD
LA_DOB	Policy_term
Marital_status	PPT
City	billing_frequency
City_classification	risk_status
STATNAME	contract_type
DSTNAME	Product_description
Focus_Region	Product_Club_Manual
Owner_salary	CUST_prod_cat
Occ_profile	Product_brief_category
Occupation	Par_NonPar
Occupation_group	ECS_flag
own_occupation	channel_flag
Freq (CustSegList)	Med_flag
multi_cust (CustSegList)	Combine_policy (CustSegList)

Step 2: Comparison of Datasets

	Merged	CustSegList
1. Shape	(812914,37)	(750598,33)
2. Observations	One row per policy purchase	One row per customer
	Continuous variables related to personal information have same values in both data sets ex- income, age	
	policy_owner_number is unique and identifies the client	
	Variables related to Insurance policies change in CustSegList: 1. sum_assured and afyp are aggregated. 2. PPT, policy term, billing frequency, max of two values is chosen 3. premium, RCD smaller of the values is chosen for majority rows	
	Categorical values remain same across both data sets except for variables with policy description	

Outcome: **Merged** data set is more relevant for further use since aggregate values present in CustSegList will not be beneficial for predictions

Step 3: Create raw data to do further Analysis

- All variables from merged were chosen
- Created variable 'age': derived from LA_DOB
- Created variable 'Freq': using count of policy_owner_number
- Created the 'target' variable:
  - Count of policy_owner_number (Freq)>1 : 1
  - Count of policy_owner_number (Freq)=1: 0
- Drop columns
  - Own_Education: abbreviated terms from Own_Edu
  - own_occupation: abbreviated terms from Occupation

Step 4: Data cleaning and treatment

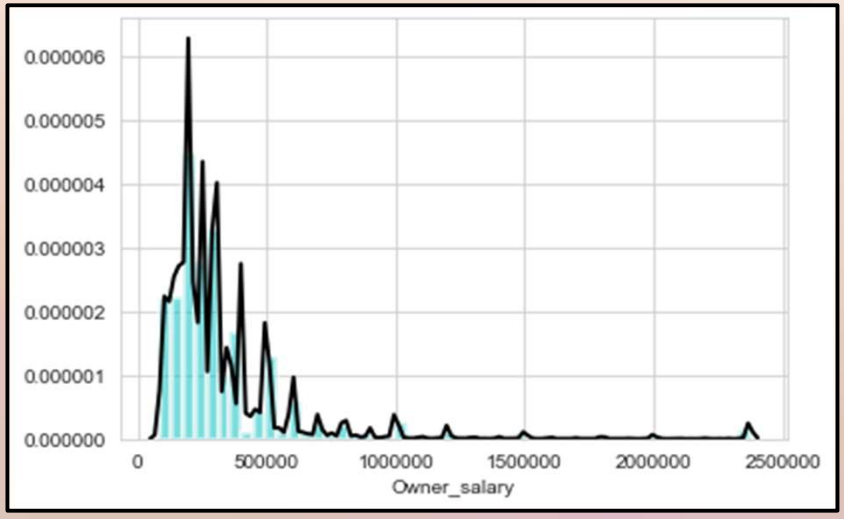
- Cleaning the Data – Cleaning tasks like converting Data Types, converting 'MISSING', 'N.A.' to NaN, String reformatting

Continous Variable

- Treating missing values for continuous variables (Owner_salary) by replacing with mean value

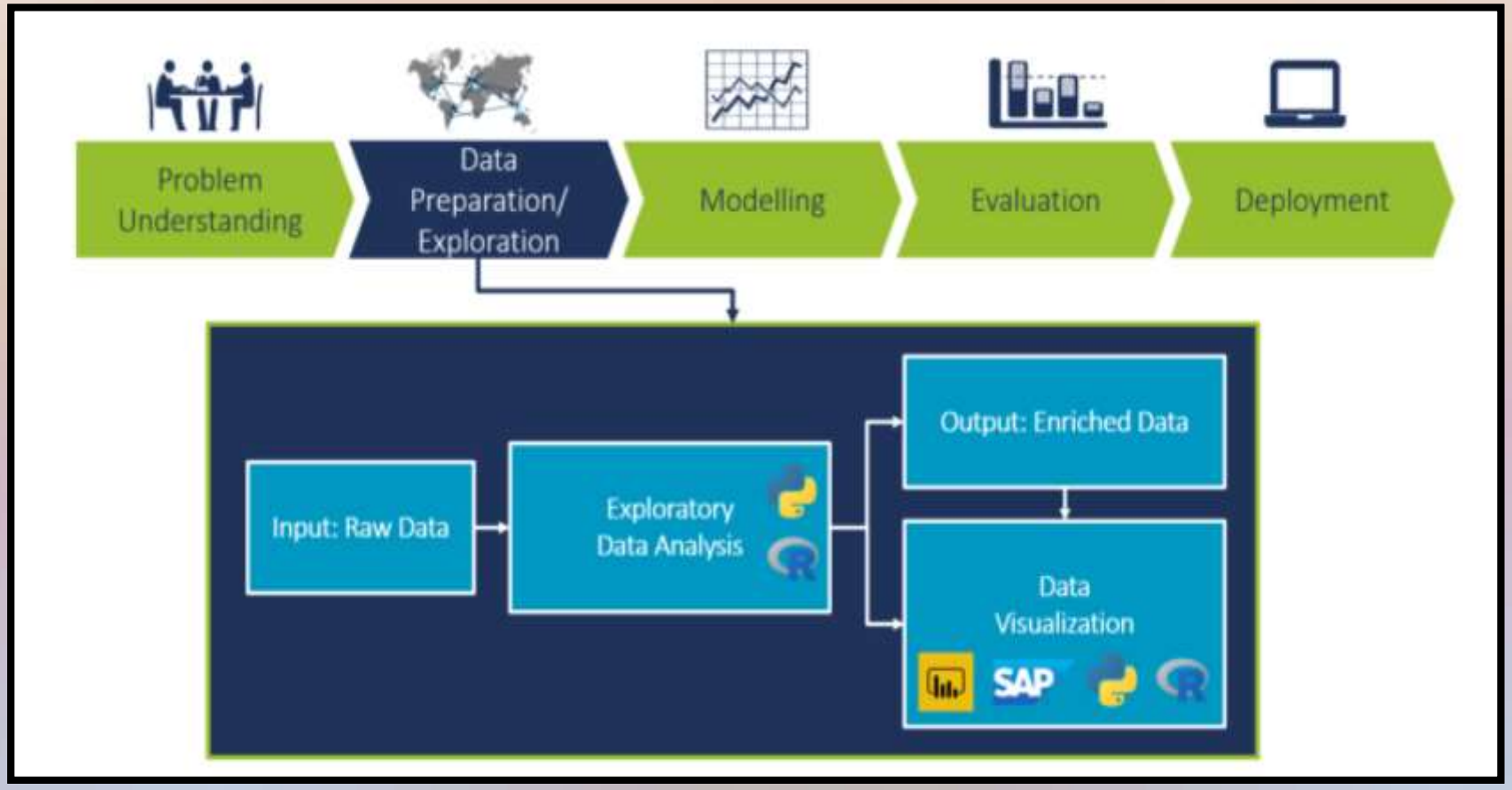
Categorical Variable

- Dropping rest of missing values (i.e., not treating categorical variables)




```
Owner_salary 1.0766  %missing values
Marital_status 0.085  %missing values
Own_Education 9.3519  %missing values
Own_Edu 9.3519  %missing values
Own_gender 0.0846  %missing values
own_occupation 0.1856  %missing values
Occupation 0.2722  %missing values
Occupation_Group 0.2722  %missing values
Focus_region 0.0608  %missing values
Occ_Profile 0.2722  %missing values
DSTNAME 0.0015  %missing values
STATNAME 0.0015  %missing values
City_classification 0.0015  %missing values
```

Exploratory Data Analysis (EDA)



Dataset for Model Building: Concept

Customer	Policy Bought
Customer 1	**A**
Customer 1	B
Customer 1	C
Customer 2	**A**
Customer 2	D
Customer 3	**A**
Customer 3	C
Customer 3	D
Customer 4	B
Customer 5	C

- As seen from above example, some of the Customers have bought multiple policies. So we sorted the data as per the 1st policy (based on RCD)
- Based on first policy, we predicted Customer buying multiple policies

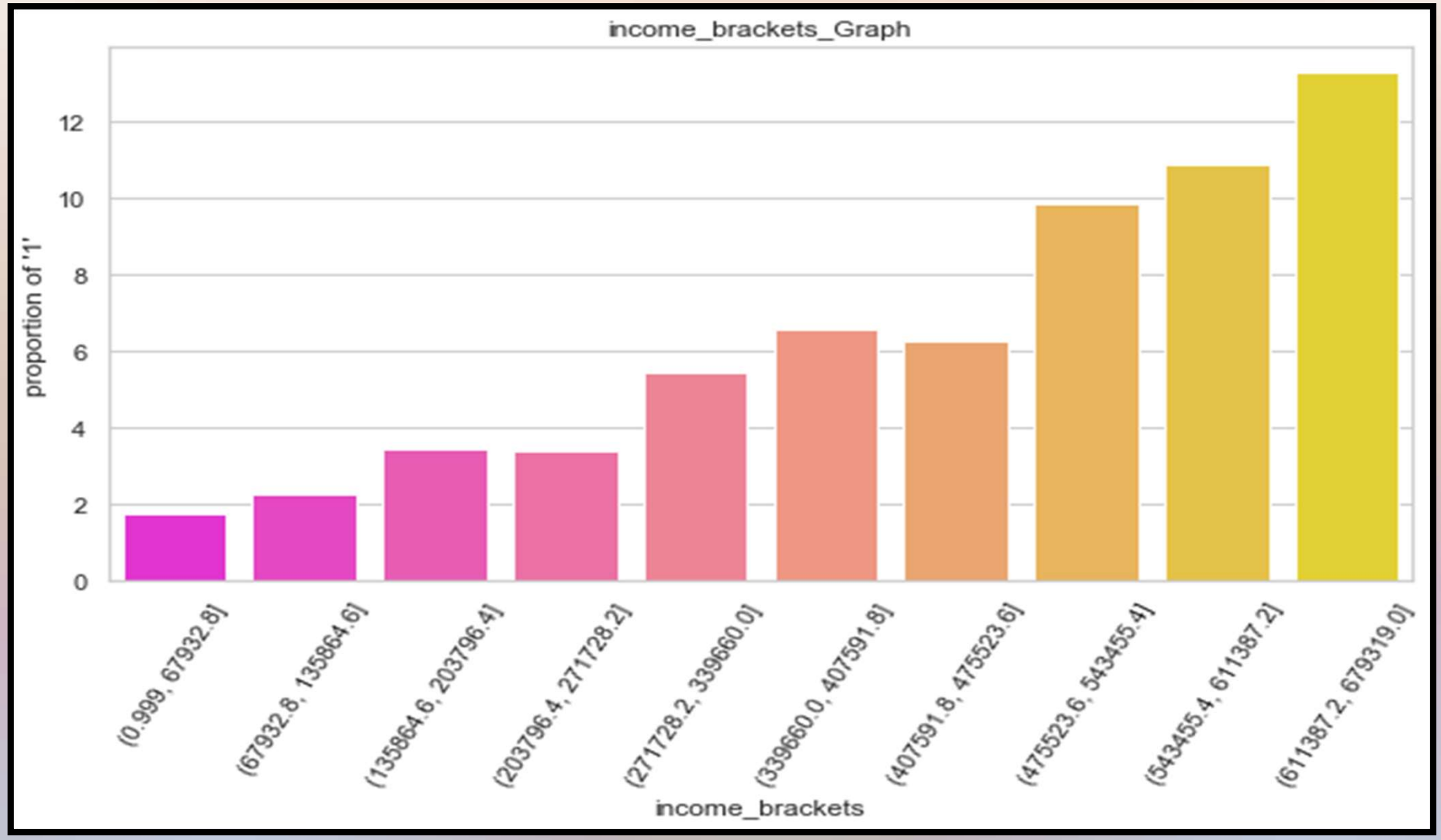
Dataset for Model Building: Execution

- Variable RCD (Risk Commencement Date) identified to fetch first policy bought by a customer

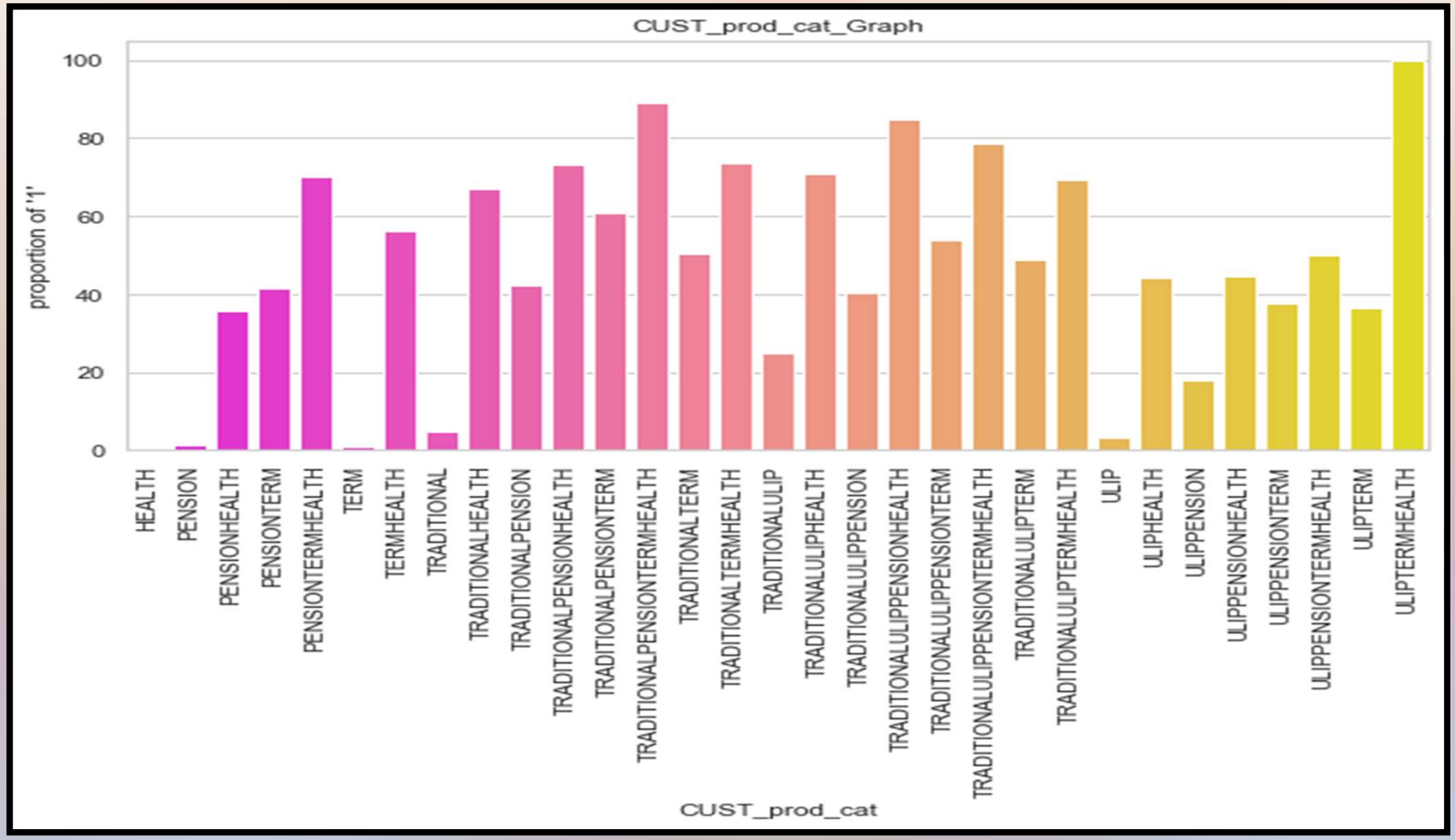
```
#create dataset to model on
df_merged.sort_values(['policy_owner_number', 'RCD'], inplace=True)
df_merged.reset_index(drop=True, inplace=True)
dataset=df_merged.drop_duplicates(subset='policy_owner_number', keep='first')
```

- Dropping columns not needed for model development
  - policy_number, policy_owner_number: Identifiers
  - RCD: sorted already
  - Freq: Target variable has been obtained

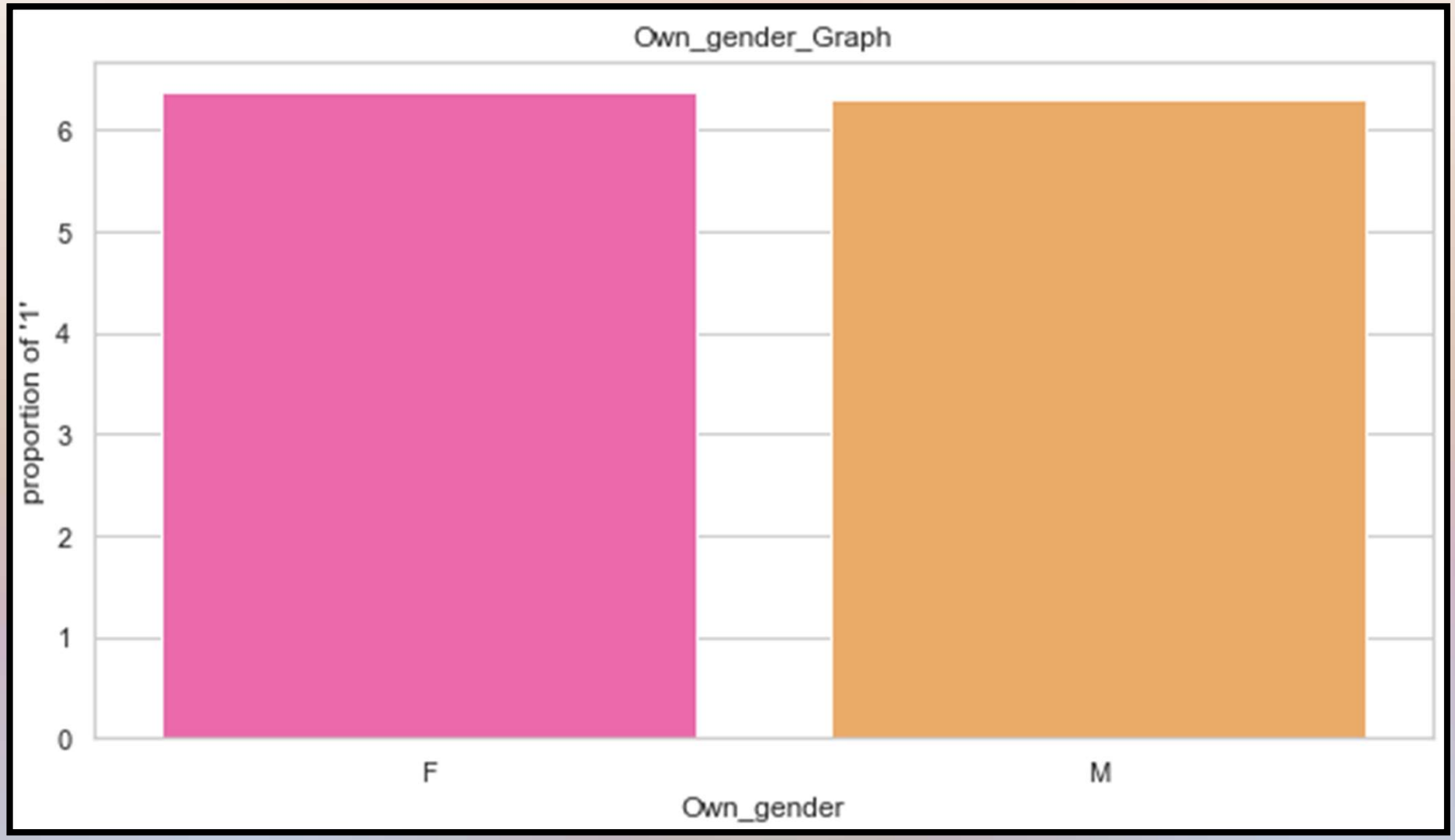
Owner Salary (significant)



Customer Product Category (significant)



Gender-wise classification (insignificant)

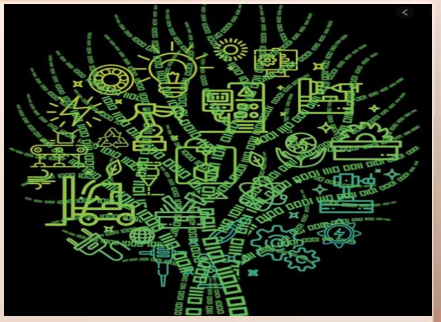


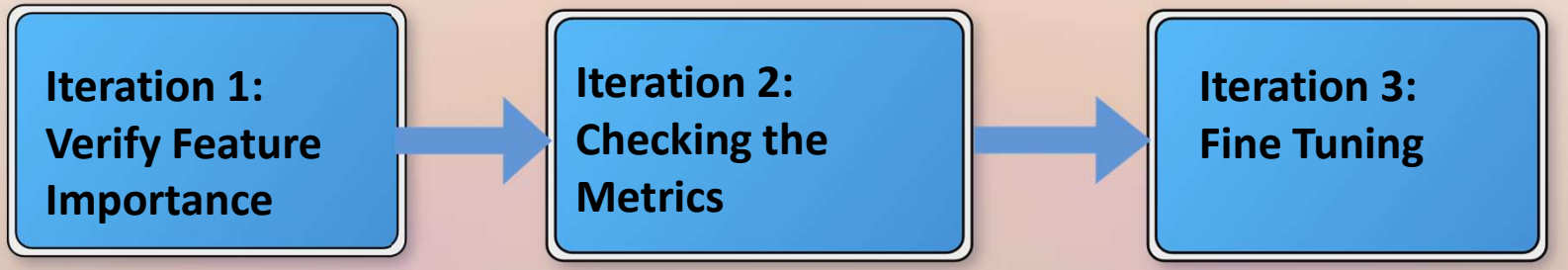
Summary of EDA

Sr. No.	Variable Category	No. of Variables	Variable Name	Impact on Target Variable
1	Amount	4	Owner_Salary, afyp, premium, sum_assured	Significant
2	Gender	2	Own_gender, LA_gender	Low
3	Education	1	Own_Edu	Moderate
4	Occupation	3	Occ_Profile, Occupation_Group, Occupation	Significant
5	Internal Categorization	2	risk_status, contract_type	Moderate
6	Product	5	Product_Description, Par_NonPar, Product_brief_Category, Product_Club_Manual, CUST_prod_cat	Significant
7	Location	5	city, DSTNAME, STATNAME, Focus_region, City_classification	Significant
8	Time	4	Age, PPT, Policy_term, billing_frequency	Significant
9	Flags	3	channel_flag, Med_Flag, ECS_flag	Moderate
10	Marital Status	1	Martial_status	Moderate
11	Identifiers	2	policy_number, policy_owner_number	NA
12	Date	1	RCD	NA
13	Frequency	2	Freq, Target	NA

Model Building

Random Forest Model



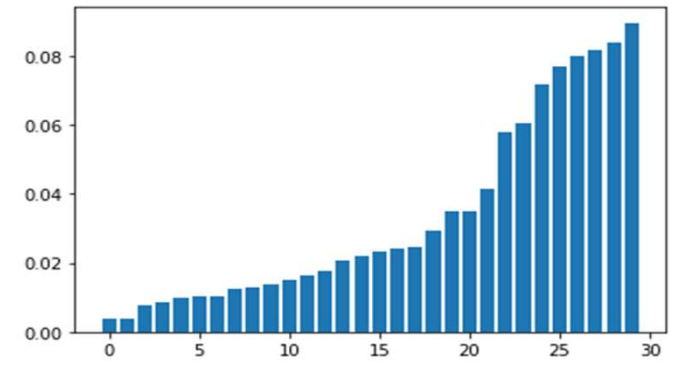


Steps:

- Feature Engineering
- Verifying important variables from iteration 1
- Checking metrics after choosing important variables from iteration 2
- Fine tuning the model in iteration 3

Iteration 1: Feature Importance

- **Feature engineering**:
  - Label encode categorical variables
- **Feature Importance**:
  - Selecting features greater than or close to 0.05 for further iteration
- **Features used**
  - 30 (including target)
- **Accuracy**
  - 93.92%



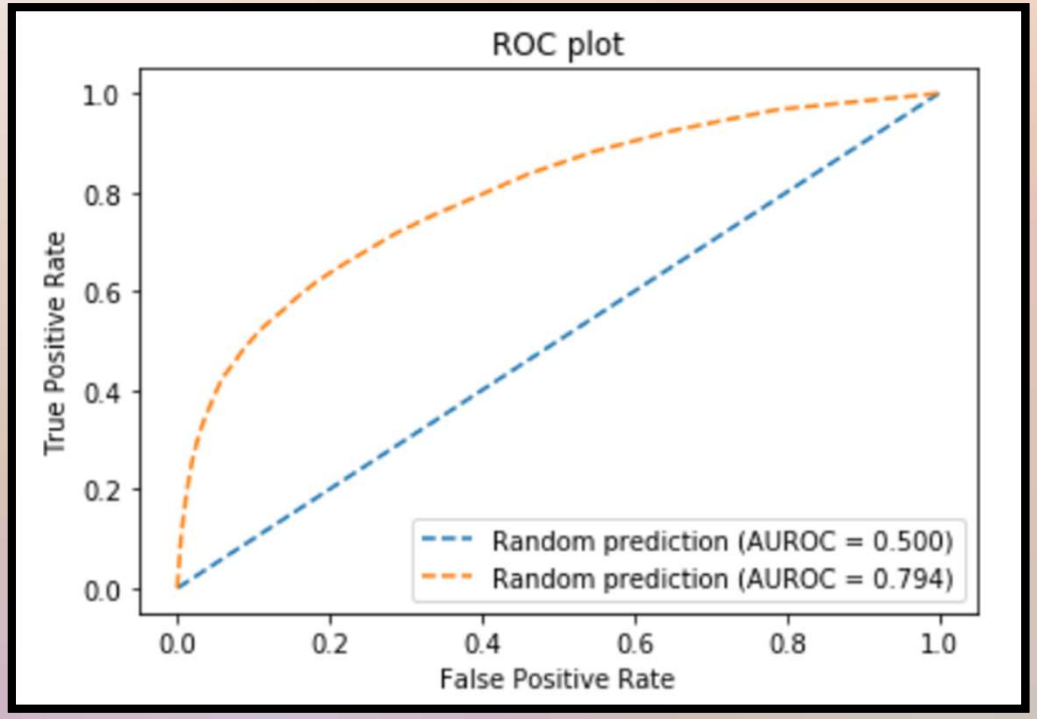

```
{'Med_Flag': 0.0036103132780933127,
 'Product_brief_category': 0.003942889966881695,
 'Par_NonPar': 0.00783010620232286,
 'billing_frequency': 0.008633024397178949,
 'LA_gender': 0.00989515060598074,
 'ECS_flag': 0.010154501350665633,
 'Own_gender': 0.010240100643347584,
 'channel_flag': 0.012270895522397903,
 'Marital_status': 0.012993972748597047,
 'Product_Club_Manual': 0.013884481997877294,
 'Occ_Profile': 0.01487766277211495,
 'Product_Description': 0.016440911196840415,
 'Focus_region': 0.017649273119928775,
 'risk_status': 0.020533235459192833,
 'Occupation_Group': 0.022011234130765167,
 'PPT': 0.023237607828855207,
 'contract_type': 0.02409288227066682,
 'City_classification': 0.024783623021481183,
 'Own_Edu': 0.029254436418316477,
 'Policy_term': 0.0348848550603813,
 'Occupation': 0.035029395203075624,
 'STATNAME': 0.041366717364934656,
 'city': 0.05789232222330059,
 'DSTNAME': 0.06062141998036463,
 'Owner_salary': 0.07159908584080715,
 'afyp': 0.07702363060510536,
 'premium': 0.07992699849148813,
 'CUST_prod_cat': 0.08174333454685098,
 'age': 0.08388140312196837,
 'sum_assured': 0.08969453463021843}
```

Iteration 2: Fitting the Model

- **Features selected**: 16 (including target)
- **Accuracy on validation data**:
  - 93.88%
- **Confusion Matrix**: We are fine with false positives, but false negatives impact us more → resources spent will not be useful for these percentage of customers

	Predicted	
Actual	0	1
0	True Negative (94.47%)	False Positive (44.16%)
1	False Negative (5.53%)	True Positive (55.84%)

Iteration 2: ROC Plot



AUC : 0.792

As observed from the graph, the bend is visible between the range 0.3 to 0.5

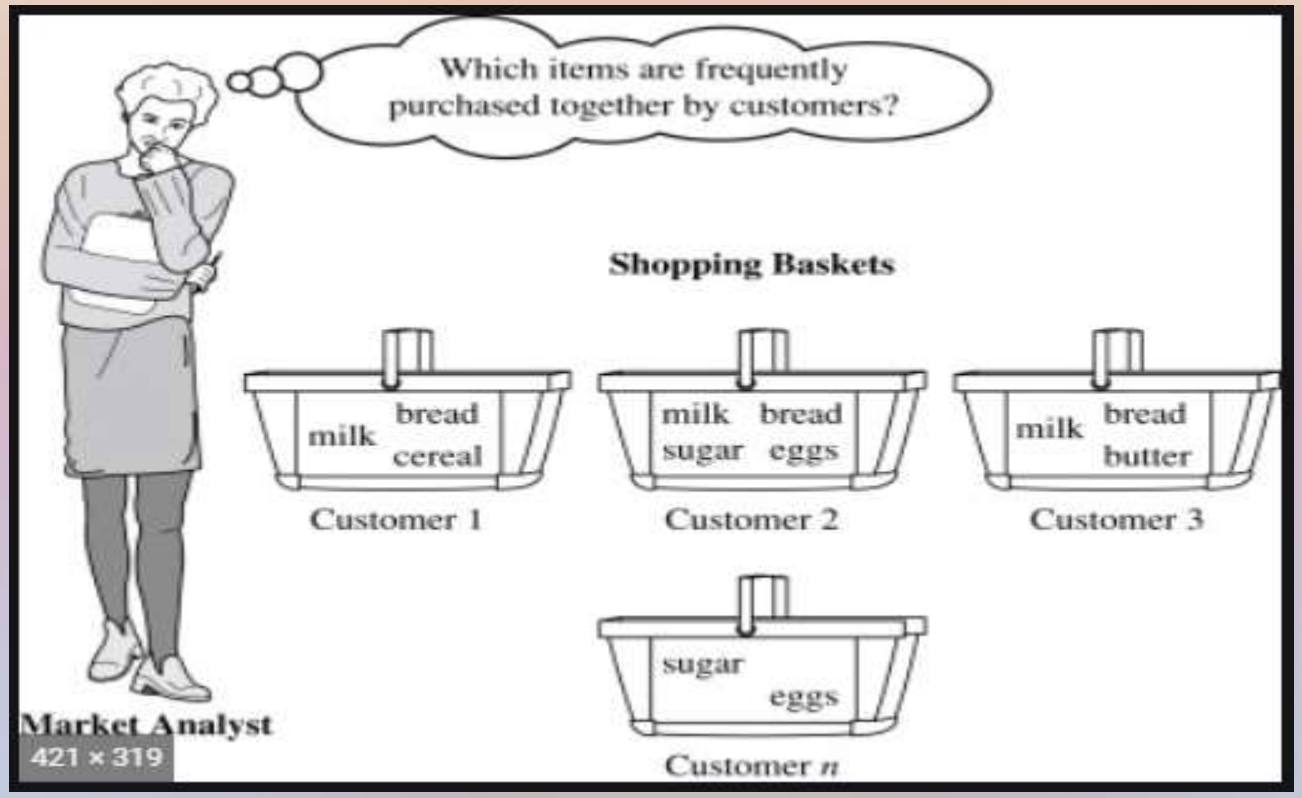
Iteration 3: Fine Tuning the Model

	Predicted	
Actual	0	1
0	True Negative (95.01%)	False Positive (51.24%)
1	False Negative (4.99%)	True Positive (48.76%)

- **Accuracy on Testing Data** : 93.63%
- **Threshold** = 0.40

- We changed the Threshold to get the False Negatives below 5% assuming the same would be tolerated by Business Team

Market Basket Analysis



Dataset for MBA: Concept

Customer	Policy Bought
Customer 1	A
Customer 1	C
Customer 2	A
Customer 2	D
Customer 3	A
Customer 3	D
Customer 3	C

- Amongst people buying multiple policies, majority buy D as a second policy
- Apriori generates such Association Rules (A → D)

Dataset for MBA: Execution

- We use all records of individual policies pertaining to a customer
- In above figure, each row indicates a customer buying multiple policies; policies chosen form the individual columns
- We noted whether a customer buys a policy (listed as 1, 0 otherwise)

Association Rules: Application

- Apriori generates rules which can be summarized as shown in table below
- Displaying rules for few Top (71%) categories (Variable: Product_Club_Manual)
- Depending on needs of the Business, more rules can be generated
- Metric 'Lift': determines how likely the recommended product will be purchased, given the product already purchased

Lift value = 1	No relation between Product A and B
Lift value > 1	Product B is likely to be bought
Lift value < 1	Product B is likely to be avoided

Conclusion

- Random Forest Model helped us to predict -
  - 'Which Customer would buy an additional policy' with an accuracy of 93.63%
- Apriori Model (Market Basket Analysis) helped us find the trend of -
  - 'Which policy the Customers would likely buy as their next policy during cross-selling'

Thank You