USE CASE STUDY REPORT

Group No. : **09**

Student Names: Rushi Prashant Chaudhari and Jayachandra Srinivas Chimakurthi

Executive Summary:

Santander Bank (Spain's) offers their customers with customized product recommendations every now and then in order to meet their needs and satisfaction and make better business. The objective of the project was to analyze 18 months of Santander Spain's customer behavior data and predict what new products the customer buys in June 2016 based on their past behavior. With the predicted data, Santander can promote their products to all the customers and can even target promotions to do better business and ensure customer gratification.

I. Background and Introduction

Santander Bank, formerly known as Sovereign Bank, is a subdivision of the Spanish Santander Group. The retail banking company is one of the largest commercial banks in the United States of America, with over 600 branches, 2000 ATMs, 9900 employees, and 2 million customers. It provides a wide range of products like credit cards, insurance, loans, debit cards, electronic banking etc. It used to (before the release of the dataset) give a wide range of product recommendations only to some of its customers. In contrast, other customers rarely got any. From the scope of the offers, only a few of them were more likely to be used by the customer, which resulted in uneven customer experience.

II. Data Preparation and Preprocessing

Data Description:

The data is extracted from the Kaggle data science competition challenge (Santander product recommendation). The data set contains artificial **Santander Spain's anonymized customer data** of 929615 test users across 24 predictors, namely between January 2015 and May 2016. All the possible products are named by the format ind_(xyz)_ult1.

Data Cleaning:

We performed several steps to clean and prepare the data for further analysis such as missing value treatments, splitting, data consolidation, character value sanitization etc.

1. Converting the Date Variable to its correct format

Transformed the dates stored in character and numeric vectors to POSIXct objects.

2. Handling Missing Data (Imputation, likewise detection, feature elimination)

Likewise Detection: -

We have applied likewise detection for handling missing values as the missing data is limited to a small number of observations.

3. Feature Elimination

We have dropped variables (ult_fec_cli_1t -> Last date as primary customer and conyuemp - spouse index) as there are more than 98% of missing observations. The features (tipodom -> address type, cod_prov -> province code and segment -> customer segment as they don't make any sense since we already have country and province information for customers.

Few other columns were removed which doesn't impact the customer purchase behaviour such as indrel_1mes -> Customer type at the beginning of the month, indext -> foreigner index and indfall -> deceased index.

4. Missing Value Imputation

We have applied mode imputation for the categorical variable "tiprel_1mes" -> Customer relation type at the beginning of the month and assigned value of "A" since it is the majority status and the columns sex0-> sex, canal_entrada -> channel of entry, nomprov -> province name, we created a new level "unknown" and assigned to it.

5. Data Sanitization

The column age is plotted below, and it is sanitized as there are people below age 18 and above 100 as it misleads the learning process. The suspect samples are filtered and replaced by the median age. The values below 18 are imputed by the median between 18 and 30 and the values above 100 are imputed with median between 50 and 100.

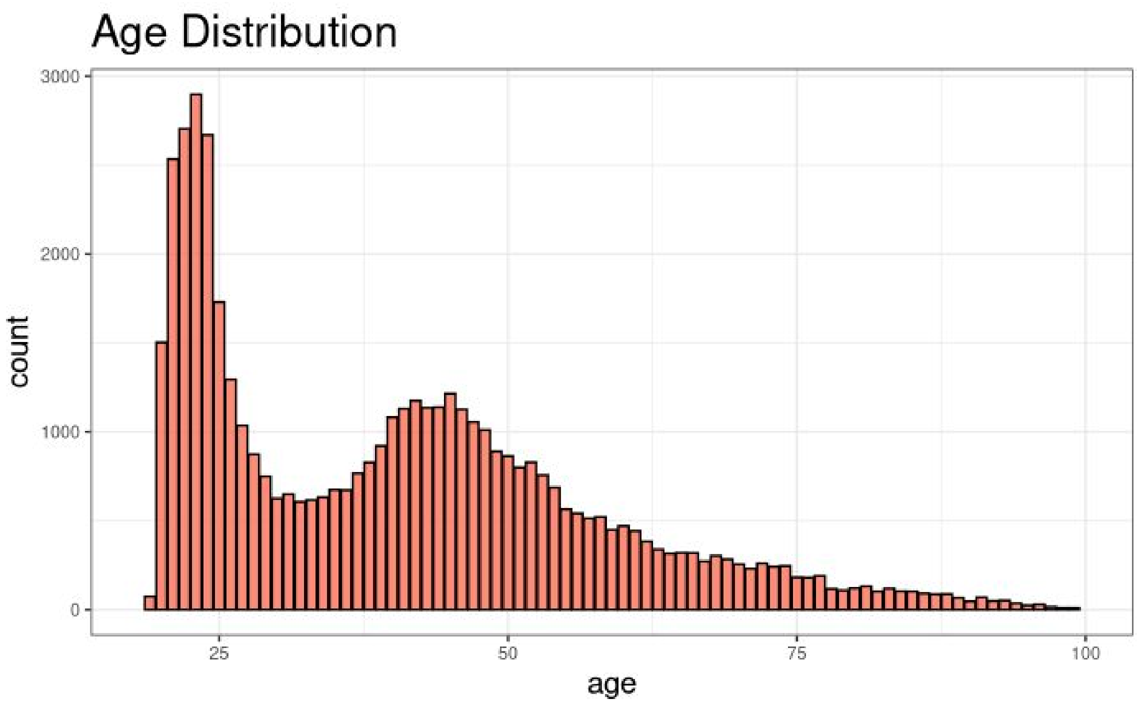

Figure 1: Age Distribution

The column "renta" (income) is imputed by median income of the city where the customer belongs as the income data is skewed.

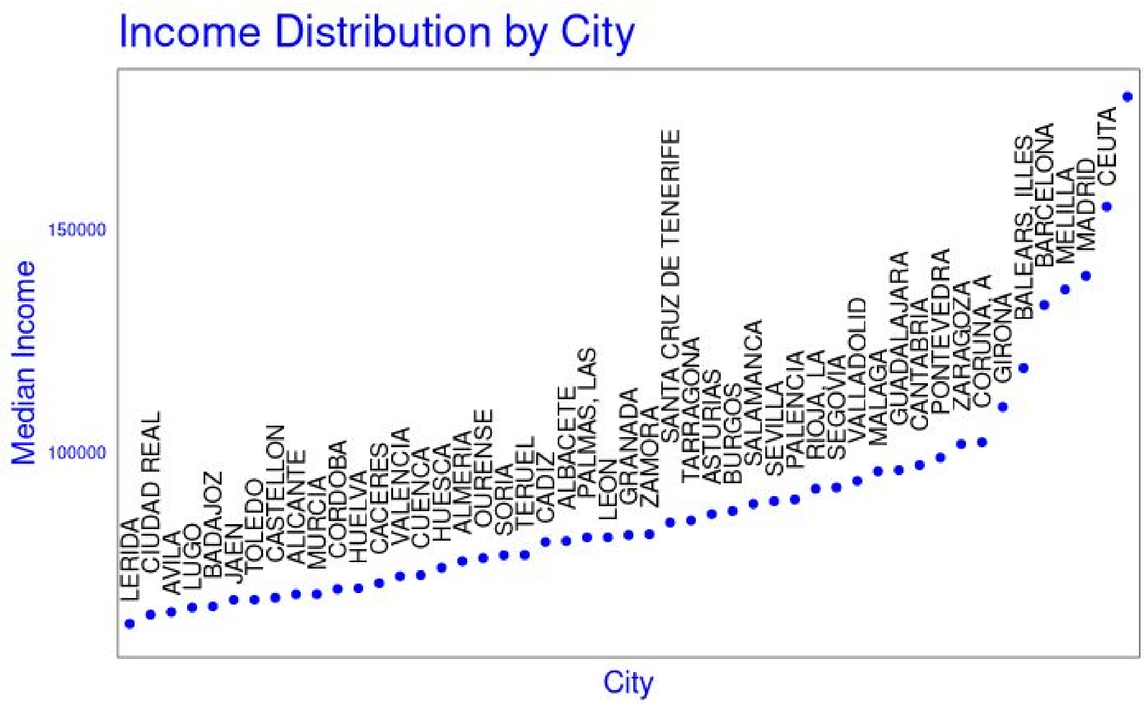

Figure 2: Income distribution versus city

III. Data Exploration and Visualization

1. Analysis of Customer Initiation into the Bank by Month: -

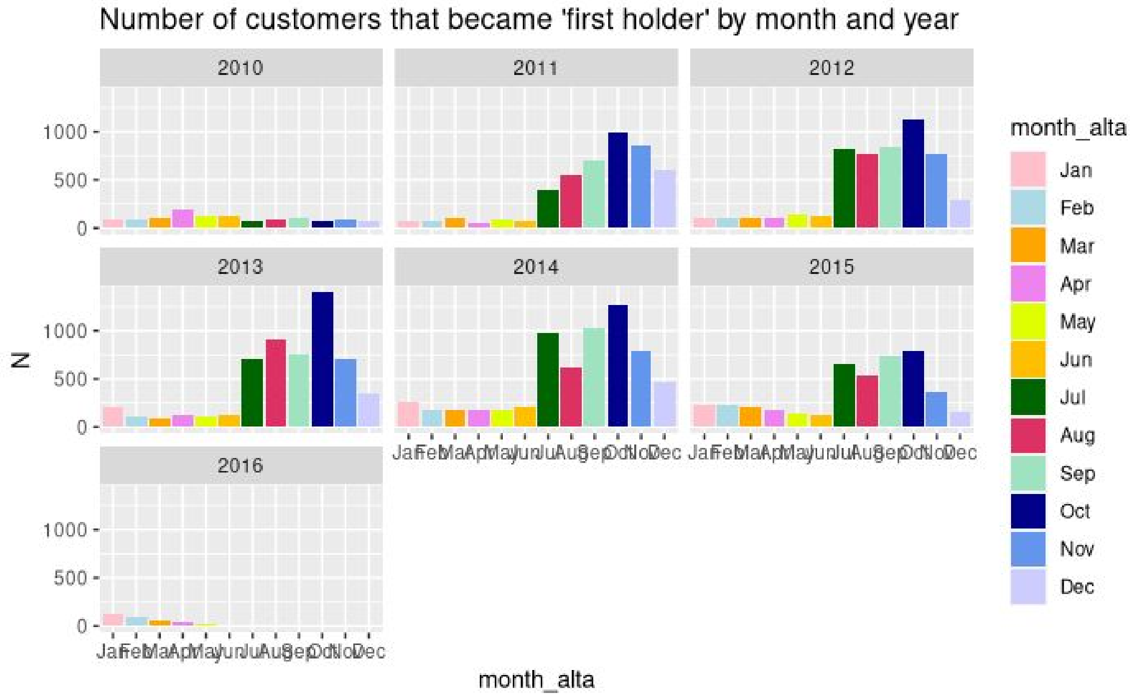

Figure 3: Customer Initiation versus Month

There is a significant rise in the number of new bank accounts in the month of July and that remains till October.

2. Analysis of Customer Age and Segmentation

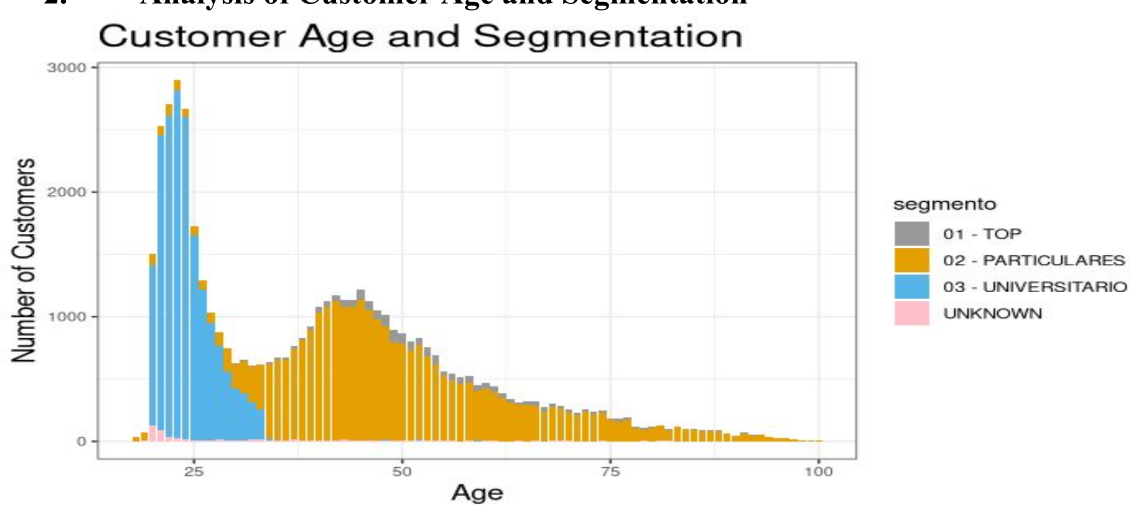

Figure 4: Customer Age versus Number of Customers

We can observe from the plot, college graduate are the young people with majority of the students fall in the age between 20-26 years and VIP and Individuals are middle aged.

3. Analysis on Channel of Entry and Household Income

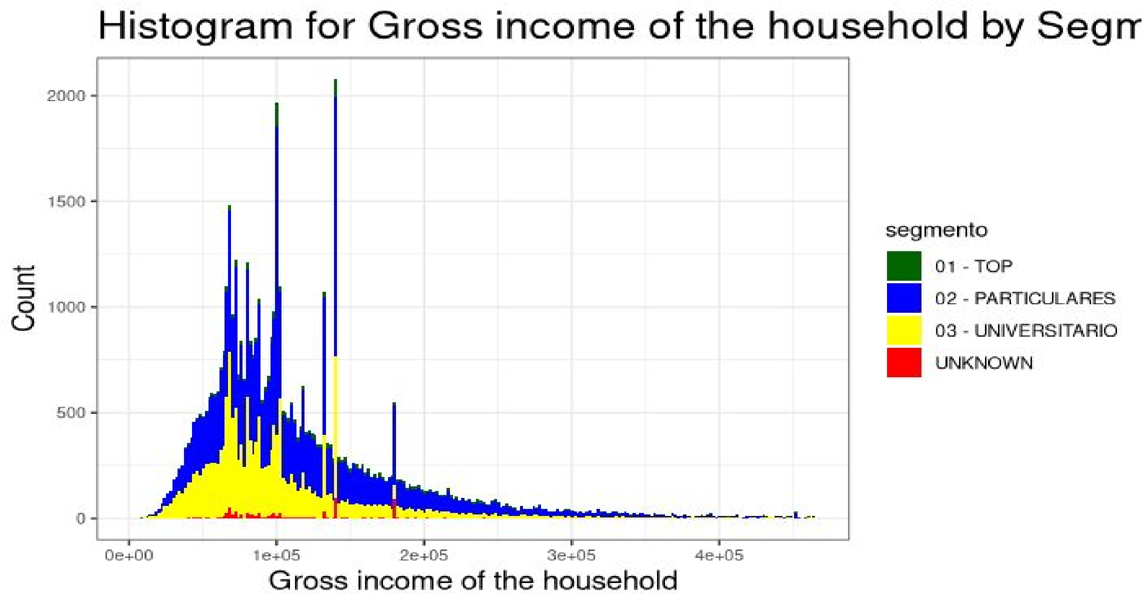

Figure 5: Gross Income by Segment.

We can observe from the plot VIP customers had the highest income and the graduate students has low income compared to household gross incomes

4. Analysis on Customer Age and Channel.

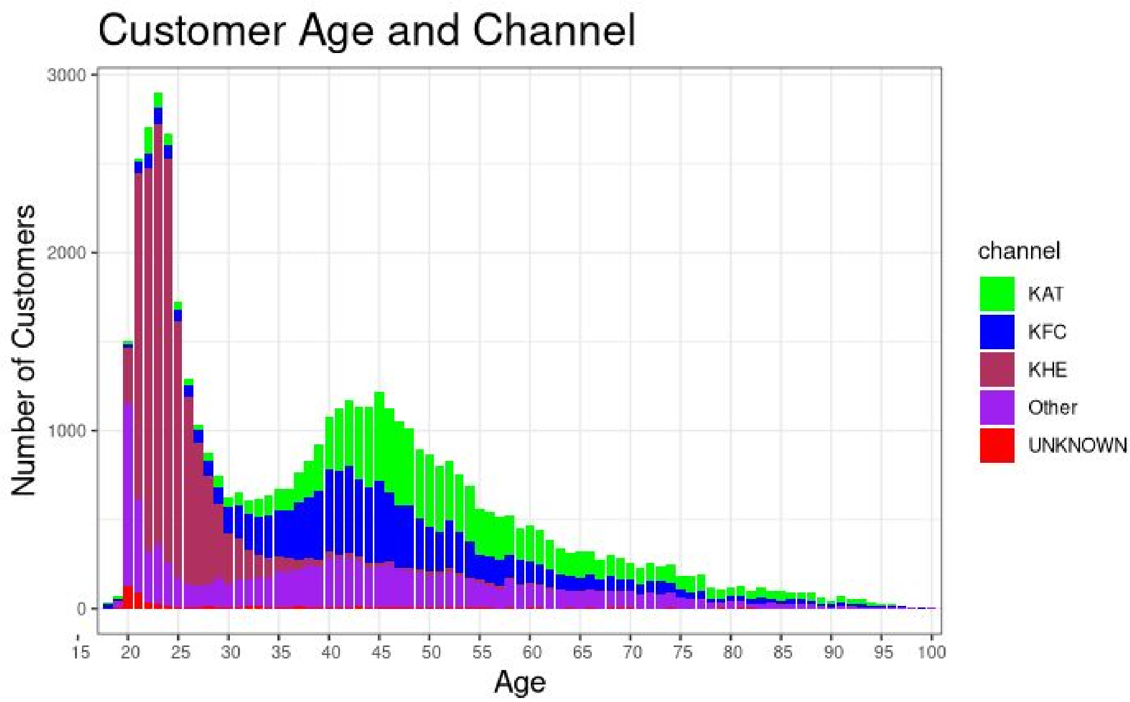

Figure 6: Customer Age versus Number of Customers across channel.

We can observe the students (18- 26 years) use "KHE" as the channel of entry.

5. Analysis on Gross Income and channel

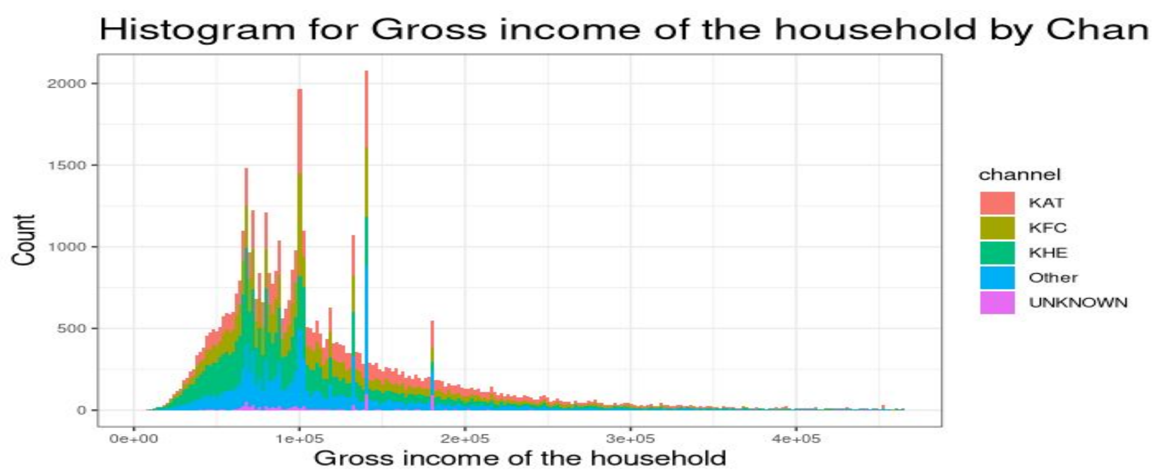

Figure 7: Gross income of household by channel.

The channel of entry is strongly correlated in a positive direction with gross household income. The customer with highest income (VIP) had entered via the KAT channel and the customer with low income (the students) had entered via the KHE channel and the customers with less income than the university students had entered the KFC and KAT channel.

6. Analysis of the Popularity of Products

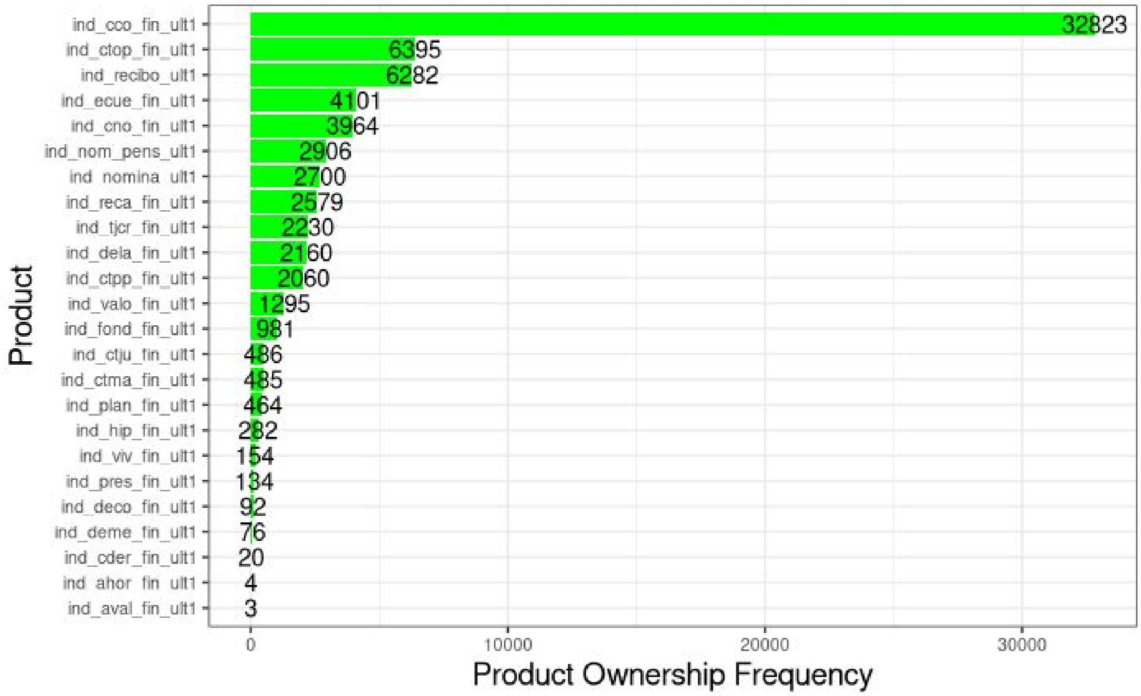

Figure 8: Product Ownership Frequency

7. Analysis on Product Popularity by Segments

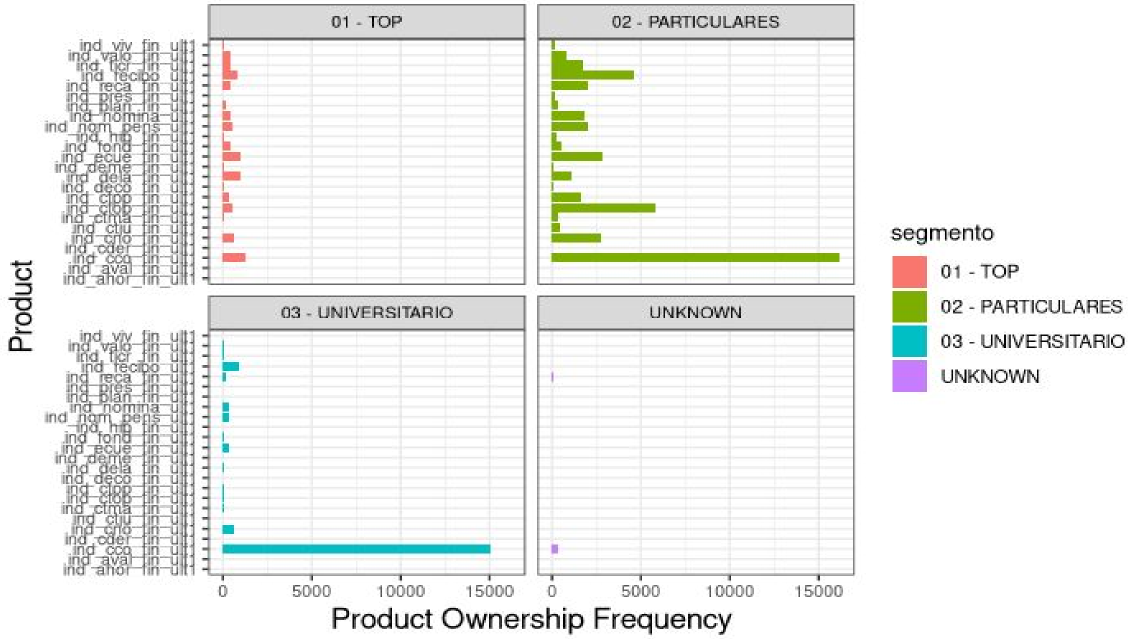

Figure 9: Product Ownership Frequency by product and Segments

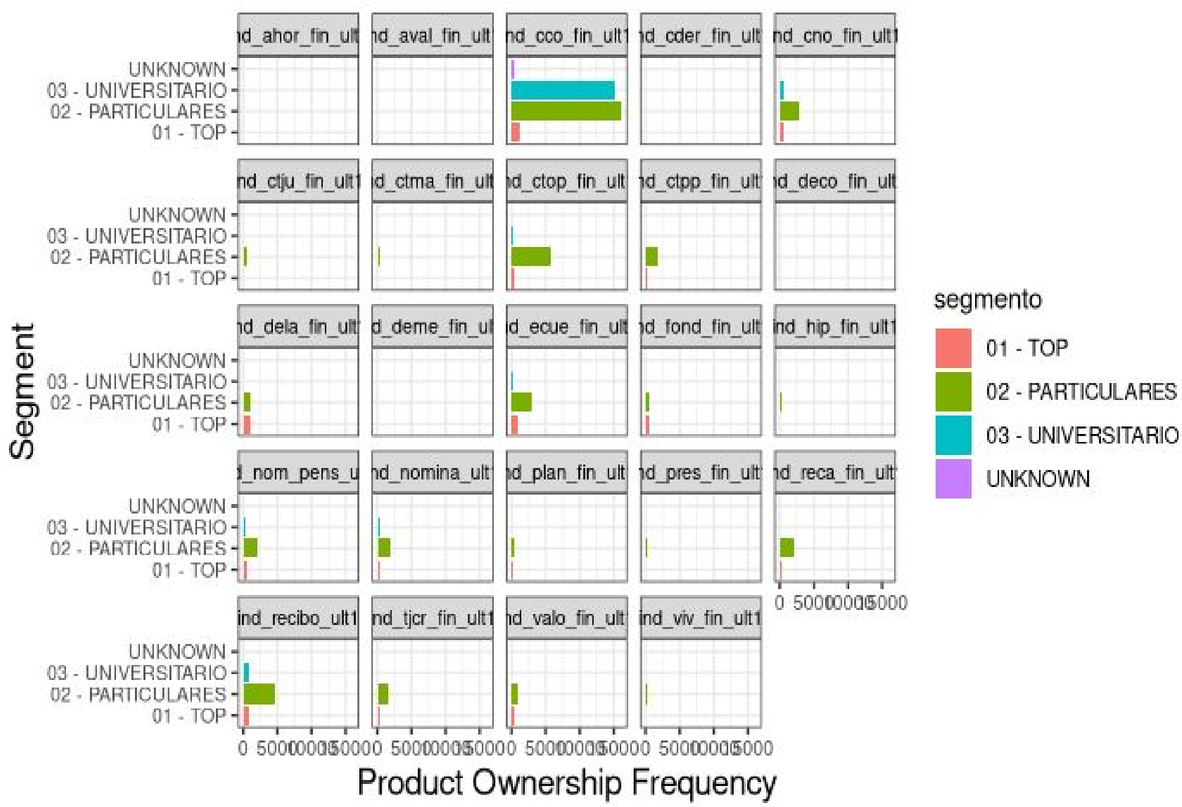

Figure 10: Product Ownership Frequency by Segments

We can observe that current accounts products are the most popular when compared to the other products with university students and the particular groups where the gross income is less than the average household income.

When it comes to the top income group, the products e-account, long term deposits and direct deposit are also popular along with current accounts. In the middle income category, the direct debit, payroll and particular products are also popular with particular groups and the next is short term deposits product.

8. Analysis on Product popularity by Sex

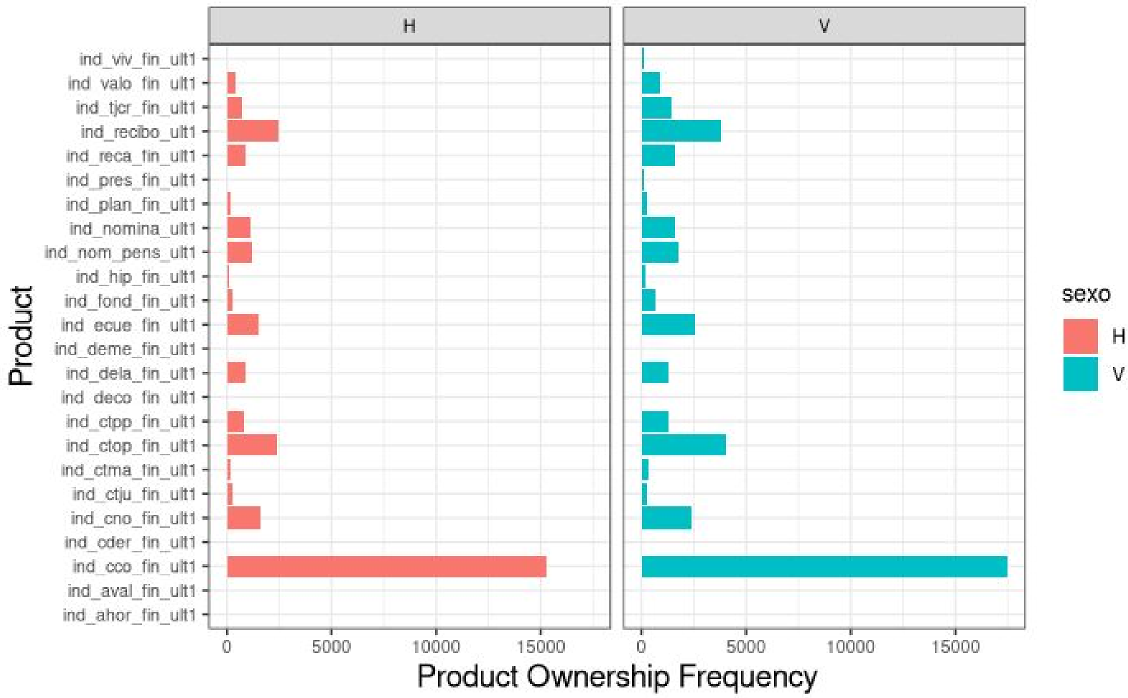

Figure 11: Product Ownership Frequency by Product and sex.

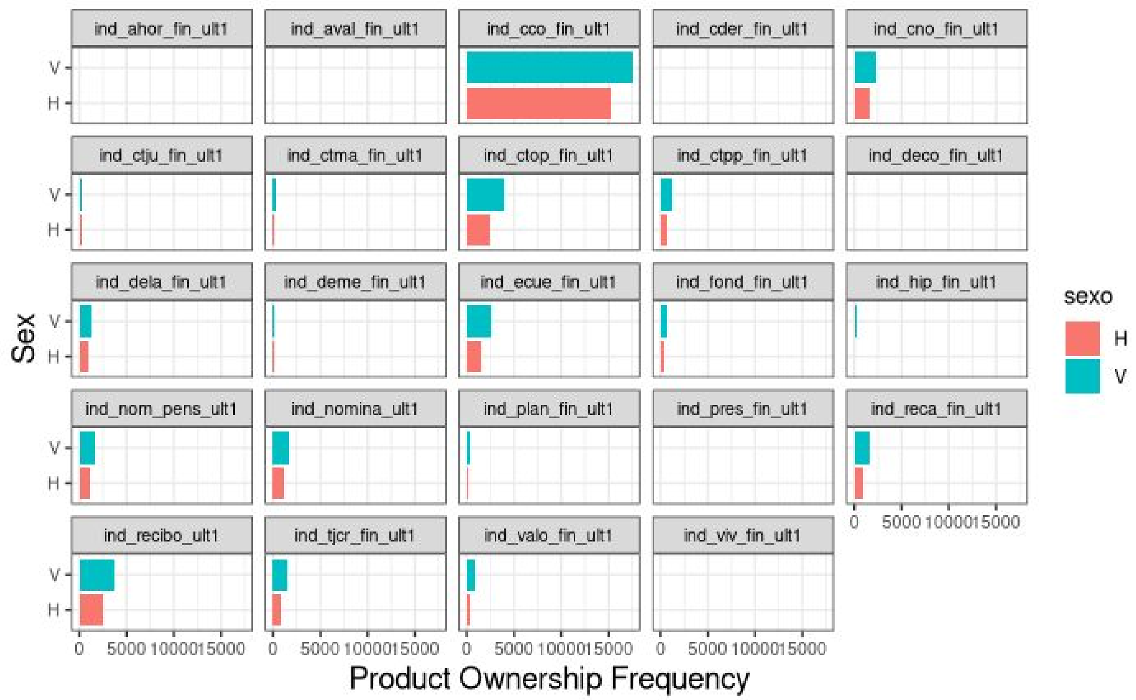

Figure 12: Product Ownership Frequency by sex.

Males are higher proportion than females in the data and there is not much difference between the products owned by females and males.

9. Association Rule Mining

Association rule mining is a procedure which aims to observe frequently occurring patterns, correlations, or associations from datasets. The Apriori Algorithm is a popular algorithm to Find Out Frequent Itemsets in Data Mining. Support, Confidence and Lift are three measures that are used to decide the relative strengths of the rules.

Consider, the rule X => Y,

$$Support(\{X\} \rightarrow \{Y\}) = \frac{Transactions\ containing\ both\ X\ and\ Y}{Total\ number\ of\ transactions}$$

$$Confidence(\{X\} \rightarrow \{Y\}) = \frac{Transactions\ containing\ both\ X\ and\ Y}{Transactions\ containing\ X}$$

$$Lift(\{X\} \rightarrow \{Y\}) = \frac{(Transactions\ containing\ both\ X\ and\ Y)/(Transactions\ containing\ X)}{Fraction\ of\ transactions\ containing\ Y}$$

Lift says how likely item Y is purchased when item X is purchased, while controlling for how popular item Y is.

A confidence of 1 implies that whenever an LHS item was purchased, the RHS item was purchased 100% of the time.

lhs	rhs	support	confidence	coverage	lift	count
{ind_nom_pens_ult1,ind_recibo_ult1}	{ind_nomina_ult1}	0.05238661	0.931205	0.056257	13.9656	20710
{ind_cno_fin_ult1,ind_nom_pens_ult1}	{ind_nomina_ult1}	0.06306124	0.9295302	0.067842	13.94049	24930
{ind_nomina_ult1}	{ind_nom_pens_ult1}	0.06667847	1	0.066678	13.93479	26360
{ind_nom_pens_ult1}	{ind_nomina_ult1}	0.06667847	0.9291505	0.071763	13.93479	26360
{ind_cno_fin_ult1,ind_nomina_ult1}	{ind_nom_pens_ult1}	0.06306124	1	0.063061	13.93479	24930
{ind_nomina_ult1,ind_recibo_ult1}	{ind_nom_pens_ult1}	0.05238661	1	0.052387	13.93479	20710
{ind_nom_pens_ult1,ind_recibo_ult1}	{ind_cno_fin_ult1}	0.0535502	0.9518885	0.056257	9.718752	21170
{ind_nomina_ult1}	{ind_cno_fin_ult1}	0.06306124	0.9457511	0.066678	9.65609	24930
{ind_nom_pens_ult1,ind_nomina_ult1}	{ind_cno_fin_ult1}	0.06306124	0.9457511	0.066678	9.65609	24930
{ind_nom_pens_ult1}	{ind_cno_fin_ult1}	0.06784206	0.9453648	0.071763	9.652146	26820

Table 1: Association Rule

We can infer that, the product combinations of {Payroll, Pensions}, {Payroll Account + Payroll , Pensions} and {Payroll + Direct Debit , Pensions}have always been bought together. From the association rules, it is clear that the Payroll, Payroll Account, Pensions, Direct Debit and Current accounts products are most likely to be purchased together.

IV. Data Mining Techniques and Implementation

We use the following algorithms on the data for the analysis: -

1. Binary Relevance Problem Transformation Method
    a. Decision Trees
    b. Logistic Regression
2. Random Forests Algorithm Adaptation Method
3. Neural Networks Algorithm Adaptation Method

V. Performance Evaluation:

We applied the above-mentioned algorithms on the Santander dataset and evaluated them using performance metrics such as Accuracy, Ham Loss and F1 score on the validation set.

The methods for multi-label classification can be grouped into two main categories (Multi Label Classification - An Overview) -

a) Problem Transformation Methods - Methods that transform the multi-label transformation problem into one or more single-label classification problems

b) Algorithm Adaptation Methods - Methods that extend specific learning algorithms to handle multi-label data directly

Evaluation Metrics for Multi Label Classification Problems -

a) Hamming Loss - Hamming Loss measures how many times on average, the relevance of an example to a class label is incorrectly predicted. It takes into account the prediction error(an incorrect label predicted) and the missing error(a relevant label not predicted), normalized over total number of classes and total number of examples.

b) Accuracy - Accuracy for each instance is defined as the proportion of the predicted correct labels to the total number (predicted and true) of labels for that instance. Overall Accuracy is the average across all instances.

c) Precision - Precision is the proportion of predicted correct labels to the total number of actual labels averaged over all instances.

d) Recall - Recall is the proportion of predicted correct labels to the total number of predicted labels averaged across all instances.

e) F1-Measure - F1 measure is nothing but the harmonic mean of precision and recall.

As in a single label classification task, the higher the value of accuracy, precision, recall and F-1 score, the better the performance of the learning algorithm.

Binary Relevance: -
This technique treats each label as a separate single class classification problem.
e.g.

X	Y_1	Y_2	Y_3	Y_4
$x^{(1)}$	0	1	1	0
$x^{(2)}$	1	0	0	0
$x^{(3)}$	0	1	0	0
$x^{(4)}$	1	0	0	1
$x^{(5)}$	0	0	0	1

Figure 13: Binary Relevance(a).

In binary relevance, this problem is broken into 4 different single class classification problems as shown in the figure below.

X	Y_1
$x^{(1)}$	0
$x^{(2)}$	1
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	0

X	Y_2
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	1
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_3
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_4
$x^{(1)}$	0
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	1

Figure 14: Binary Relevance(b).

The makeMultilabelBinaryRelevanceWrapper provides its implementation in R.

ROC Curve for Binary Relevance Decision Tree Model: -

Classification and Regression Trees or CART for short refer to Decision Tree algorithms that can be used for classification or regression predictive modeling problems. The CART algorithm provides a foundation for important algorithms like bagged decision trees, random forest and boosted decision trees.

In our regression tree we have used entropy as

$$\text{Entropy} = -\sum_{i=1}^{N}(p(ci)\log p(ci) + q(ci)\log q(ci))$$

where q(ci) = 1-p(ci) and p(ci) is the relative frequency of class ci.
We achieved a mean accuracy and F1-score of 0.999
The binary performance measure for each label for decision trees are shown below.

	acc.test.mean	fpr.test.mean	tpr.test.mean	auc.test.mean
ind_ahor_fin_ult1	0.9999326	0.0000000000	0.00000000	0.5000000
ind_aval_fin_ult1	1.0000000	0.0000000000	NaN	NA
ind_cco_fin_ult1	0.7411876	0.4767096135	0.85345180	0.7230919
ind_cder_fin_ult1	0.9995956	0.0000000000	0.00000000	0.5000000
ind_cno_fin_ult1	0.9178405	0.0000000000	0.00000000	0.5000000
ind_ctju_fin_ult1	0.9995956	0.0002025795	0.89285714	0.9285316
ind_ctma_fin_ult1	0.9901597	0.0000000000	0.00000000	0.5000000
ind_ctop_fin_ult1	0.8857586	0.0415184784	0.39148713	0.8801761
ind_ctpp_fin_ult1	0.9590888	0.0000000000	0.00000000	0.5000000
ind_deco_fin_ult1	0.9981128	0.0000000000	0.00000000	0.5000000
ind_deme_fin_ult1	0.9983824	0.0000000000	0.00000000	0.5000000
ind_dela_fin_ult1	0.9578082	0.0079768460	0.23546945	0.7131209
ind_ecue_fin_ult1	0.9175709	0.0033840948	0.05385852	0.6064179
ind_fond_fin_ult1	0.9788367	0.0000000000	0.00000000	0.5000000
ind_hip_fin_ult1	0.9937993	0.0000000000	0.00000000	0.5000000
ind_plan_fin_ult1	0.9917773	0.0000000000	0.00000000	0.5000000
ind_pres_fin_ult1	0.9972366	0.0000000000	0.00000000	0.5000000
ind_reca_fin_ult1	0.9481701	0.0000000000	0.00000000	0.5000000
ind_tjcr_fin_ult1	0.9532924	0.0000000000	0.00000000	0.5000000
ind_valo_fin_ult1	0.9723664	0.0000000000	0.00000000	0.5000000
ind_viv_fin_ult1	0.9970344	0.0000000000	0.00000000	0.5000000
ind_nomina_ult1	0.9442610	0.0000000000	0.00000000	0.5000000
ind_nom_pens_ult1	0.9406888	0.0000000000	0.00000000	0.5000000
ind_recibo_ult1	0.8717396	0.0000000000	0.00000000	0.5000000

Table 2: Binary Performance measure for decision tree..

The five products ind_ctju_fin_ult1, ind_cco_fin_ult1, ind_ctop_fin_ult1, ind_dela_fin_ult1, ind_ecue_fin_ult1. Had notable AUC compared to other products. Their ROC curves are shown below.

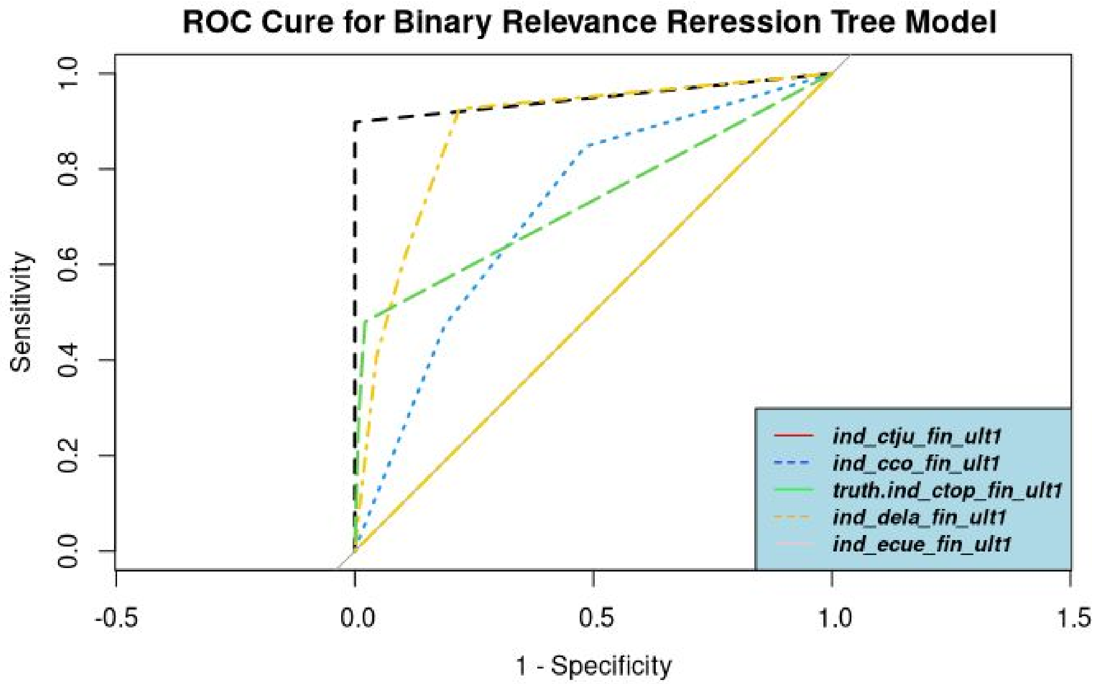

Figure 15: ROC Curve for Binary Relevance Regression Tree Model

Binary Relevance Logistic Trees: -

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). It is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

At the center of the logistic regression analysis is the task estimating the log odds of an event. Mathematically, logistic regression estimates a multiple linear regression function defined as logit(p):

$$= \log\left(\frac{p(y=1)}{1-(p=1)}\right) = \beta_0 + \beta_1 x_{i1} + \beta_2 \cdot x_{i2} + \_ + \beta_p \cdot x_{in}$$

The binary performance measure for each label for logistic regression are shown below.

	acc.test.mean	fpr.test.mean	tpr.test.mean	auc.test.mean
ind_ahor_fin_ult1	0.9997976	0.000000e+00	0.000000000	0.5000000
ind_aval_fin_ult1	0.9997976	6.747183e-05	0.000000000	0.4937926
ind_cco_fin_ult1	0.7379748	5.006051e-01	0.857881399	0.7545024
ind_cder_fin_ult1	0.9996627	0.000000e+00	0.000000000	0.8088406
ind_cno_fin_ult1	0.9199892	3.665689e-04	0.001690617	0.8658264
ind_ctju_fin_ult1	0.9995278	4.729091e-04	1.000000000	0.9998038
ind_ctma_fin_ult1	0.9910949	1.361192e-04	0.000000000	0.8651708
ind_ctop_fin_ult1	0.8577886	3.965544e-02	0.175529169	0.8904070
ind_ctpp_fin_ult1	0.9586453	0.000000e+00	0.000000000	0.8109905
ind_deco_fin_ult1	0.9979761	6.759497e-05	0.000000000	0.8389398
ind_deme_fin_ult1	0.9979087	0.000000e+00	0.000000000	0.8586638
ind_dela_fin_ult1	0.9598597	8.463817e-03	0.263565891	0.9262382
ind_ecue_fin_ult1	0.9196519	1.048618e-02	0.116357504	0.8640995
ind_fond_fin_ult1	0.9801660	2.752736e-04	0.006849315	0.9228435
ind_hip_fin_ult1	0.9942657	0.000000e+00	0.000000000	0.9305724
ind_plan_fin_ult1	0.9908925	1.361470e-04	0.000000000	0.9253290
ind_pres_fin_ult1	0.9975713	0.000000e+00	0.000000000	0.8869634
ind_reca_fin_ult1	0.9501450	0.000000e+00	0.000000000	0.8550021
ind_tjcr_fin_ult1	0.9544627	1.413428e-04	0.000000000	0.8983347
ind_valo_fin_ult1	0.9718006	1.388214e-04	0.000000000	0.9135046
ind_viv_fin_ult1	0.9964919	0.000000e+00	0.000000000	0.8962989
ind_nomina_ult1	0.9462997	2.138275e-04	0.000000000	0.8743642
ind_nom_pens_ult1	0.9419146	7.161785e-05	0.000000000	0.8681809
ind_recibo_ult1	0.8712811	4.948199e-03	0.023822128	0.8742905

Table 3: Binary Performance measure for Logistic Regression.

Random Forests Algorithm Adaptation Method: -

The Random Forests Algorithm Adaptation method extends the implementation of Random Forests to accommodate multi-label classification task. Here the algorithm extends the implementation so multiple labels can be used as the leaves of the tree. The package "randomForestSRC" has a similar implementation and is used for this analysis. The performance metrics for multi label random forests are shown below.

```
Model for learner.id=multilabel.randomForestSRC; learner.class=multilabel.randomForestSRC
Trained on: task.id = products_rfsrc; obs = 10500; features = 16
Hyperparameters: na.action=na.impute
Prediction: 10500 observations
predict.type: response
threshold:
time: 8.84
... (#rows: 10500, #cols: 49)
multilabel.subset01  multilabel.hamloss      multilabel.acc       multilabel.f1
timepredict
       1.523810e-03        6.746032e-05        9.984762e-01        9.984762e-01
8.837000e+00
multilabel.subset01  multilabel.hamloss      multilabel.acc       multilabel.f1
timepredict
       8.271299e-04        3.446374e-05        9.991729e-01        9.991729e-01
4.698000e+00
```

Table 4: Multi Label Performance measure for SRC Random Forest..

SRC random forest seems to be a very powerful algorithm, it can handle multiple target labels and resolve most missing value problems.

Neural Networks Algorithm Adaptation Method: -

Back Propagation Multi Label Learner with multiple outputs is an implementation of neural networks in the multi label classification space. The package "neuralnet" in R is used for this analysis. It was proposed by Zhang et al. in 2006 [1]. It is a single hidden layer, fully connected feed-forward architecture, which uses the backpropagation of error algorithm to optimise a variation of the ranking loss function
that takes pairwise label associations into account. This loss function can
be defined as follows

$$E = \sum_{i=1}^{n} \frac{1}{|\boldsymbol{y}_i||\bar{\boldsymbol{y}}_i|} \sum_{(k,l)\in(\boldsymbol{y}_i \times \bar{\boldsymbol{y}}_i)} exp(-(c_i^{(k)} - c_i^{(l)}))$$

Here y_i indicates the set of labels assigned to x_i and $\bar{y}_i$ indicates the set of labels which are not assigned to x_i.
$y^{(l)}_i = +1$ if the label is relevant to x_i and -1 if irrelevant.
$c^{(k)}_i$ and $c^{(l)}_i$ are the outputs of the k^{th} and the l^{th} output units representing the corresponding label predictions for the data point xi.
The neuralnet library in R provides its implementation. The activation function is set to Logistic. The mean Accuracy achieved by Neural Networks is 0.905.

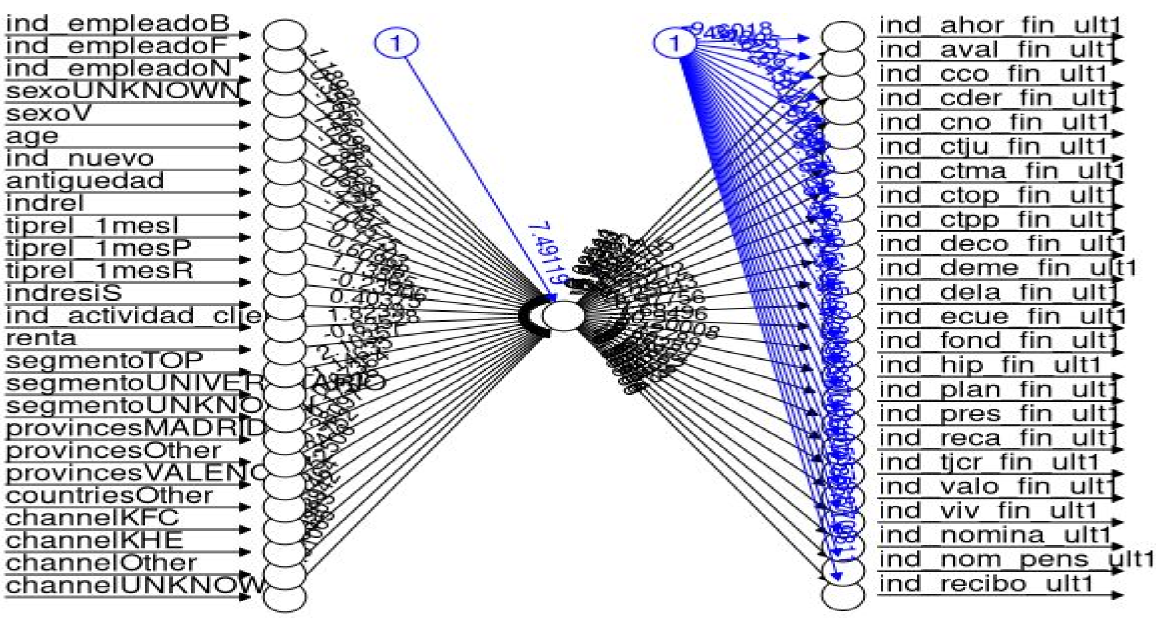

Figure 16: Neural Network Architecture

Performance on Testing data (15000 samples)

	Binary Relevance Regression Tree	Binary Relevance Logistic Tree	Random Forests	Neural Network
Performance Metrics				
Accuracy	0.6015997	0.591081	0.9991729	0.905715
Ham Loss	0.0438662	0.045039	0.0000345	
F1 Score	0.6310979	0.6229178	0.9991729	

Table 5: Performance Metrics for Model Evaluation..

Based on the results, we can see the performance of the Random Forests Algorithm Adaptation Technique is the best on this task. The Neural Network Algorithm Adaptation Technique performs well too. The performance of the learners is similar in both the test and training data which shows that the techniques are robust and are not prone to Overfitting.

VI. Discussion and Recommendation

We observe the Random Forests Algorithm Adaption Technique is best on this data, but Neural Network Algorithm Adaption Technique can perform better on further hyper tuning the parameters.

VII. Summary

In this study we test many different machine learning models to predict what new products a customer will buy based on his past behavior. We train the model on a normalized dataset and can conclude that a Random Forests Algorithm Adaption Technique performs the best in this task, and this improves the Santander Recommendation System and makes better business.

Appendix: R Code for use case study

#Importing the required packages in the work environment.

```{r message=FALSE, warning=FALSE}
library(data.table)
library(dplyr)
library(tidyr)
library(lubridate)
library(ggplot2)
library(stringr)
library(arules)
library(rattle)
library(mlr)
library(randomForestSRC)
library(rFerns)
library(neuralnet)
```

#Importing the training dataset and test dataset from Kaggle.

```{r}
data <- read.csv("train_50k.csv")
data <- data[, -c(1)]
test_data <- read.csv("/home/doraemon/test_ver2.csv")
data <- data[-c(1)]
test_data <- test_data[-c(1)]

```

#Data Cleaning:
#1. Converting the Date Variable to its correct format

```{r message=FALSE, warning=FALSE}
data$fecha_alta <- ymd(data$fecha_alta)
test_data$fecha_alta <- ymd(test_data$fecha_alta)
```

#2. Handling Missing Data (Imputation, likewise detection, feature elmination)

```
``{r}
prod_cols <- colnames(data[str_detect( colnames(data),"_ult1")])
all_cols <- colnames(data)
```

user_cols <- colnames(data[! all_cols %in% prod_cols])
data_missing_columns <- names(data)[which(sapply(data, function(x) any(is.na(x))))]
missing_data <- sapply(data[,data_missing_columns], function(x) sum(is.na(x)))
missing_data_pct <- sapply(data[,data_missing_columns], function(x)
round(sum(is.na(x))/dim(data)[1],5))
complete_cases_pct <-
dim(data[complete.cases(data[user_cols]),])[1]/dim(data[user_cols])[1]*100
```

#3. Feature Elimination

```{r}
data <- data[, !(colnames(data) %in% c("conyuemp","ult_fec_cli_1t"))]
data <- data[, !(colnames(data) %in% c("tipodom","cod_prov"))]
data <- data[,!names(data) %in% c("indrel_1mes","indext","indfall")]

test_data <- test_data[, !(colnames(test_data) %in% c("conyuemp","ult_fec_cli_1t"))]
test_data <- test_data[, !(colnames(test_data) %in% c("tipodom","cod_prov"))]
test_data <- test_data[,!names(test_data) %in% c("indrel_1mes","indext","indfall")]

data$year_alta <- year(data$fecha_alta)
data$month_alta <- month(data$fecha_alta,label=T)
test_data$year_alta <- year(test_data$fecha_alta)
test_data$month_alta <- month(test_data$fecha_alta,label=T)
```

#4. Missing Value Imputation

```{r}
data$nomprov <- as.character(data$nomprov)
data$nomprov[is.na(data$nomprov)] <- "UNKNOWN"
data$canal_entrada <- as.character(data$canal_entrada)
data$canal_entrada[is.na(data$canal_entrada)] <- "UNKNOWN"
data$sexo <- as.character(data$sexo)
data$sexo[is.na(data$sexo)] <- "UNKNOWN"
data$ind_nomina_ult1[is.na(data$ind_nomina_ult1)] <- 0
data$ind_nom_pens_ult1[is.na(data$ind_nom_pens_ult1)] <- 0
data$tiprel_1mes <- as.character(data$tiprel_1mes)

data$tiprel_1mes[is.na(data$tiprel_1mes)] <- "A"
data$segmento <- as.character(data$segmento)
data$segmento[is.na(data$segmento)] <- "UNKNOWN"

test_data$nomprov <- as.character(test_data$nomprov)
test_data$nomprov[is.na(test_data$nomprov)] <- "UNKNOWN"

````
test_data$canal_entrada <- as.character(test_data$canal_entrada)
test_data$canal_entrada[is.na(test_data$canal_entrada)] <- "UNKNOWN"
test_data$sexo <- as.character(test_data$sexo)
test_data$sexo[is.na(test_data$sexo)] <- "UNKNOWN"
test_data$tiprel_1mes <- as.character(test_data$tiprel_1mes)
test_data$tiprel_1mes[is.na(test_data$tiprel_1mes)] <- "A"
test_data$segmento <- as.character(test_data$segmento)
test_data$segmento[is.na(test_data$segmento)] <- "UNKNOWN"

all_cols <- colnames(data)
user_cols <- colnames(data[! all_cols %in% prod_cols])
data$tiprel_1mes <- as.factor(data$tiprel_1mes)
data$sexo <- as.factor(data$sexo)
data$canal_entrada <- as.factor(data$canal_entrada)
data$segmento <- as.factor(data$segmento)
data$nomprov <- as.factor(data$nomprov)

all_cols <- colnames(test_data)
user_cols <- colnames(test_data[! all_cols %in% prod_cols])
test_data$tiprel_1mes <- as.factor(test_data$tiprel_1mes)
test_data$sexo <- as.factor(test_data$sexo)
test_data$canal_entrada <- as.factor(test_data$canal_entrada)
test_data$segmento <- as.factor(test_data$segmento)
test_data$nomprov <- as.factor(test_data$nomprov)
```

```{r}
ggplot(data=data,aes(x=age)) +
  geom_bar(alpha=0.75,fill="tomato",color="black") +
  xlim(c(18,100)) +
  ggtitle("Age Distribution") +
  my_theme

data$age[(data$age < 18)] <- median(data$age[(data$age >= 18) & (data$age <=30)])
data$age[(data$age > 100)] <- median(data$age[(data$age >= 30) & (data$age <=100)])

test_data$age[(test_data$age < 18)] <- median(test_data$age[(test_data$age >= 18) &
(test_data$age <=30)])
test_data$age[(test_data$age > 100)] <- median(test_data$age[(test_data$age >= 30) &
(test_data$age <=100)])
```

```{r}
data %>%
  filter(!is.na(renta)) %>%
  group_by(nomprov) %>%
````

```
  summarise(med.income = median(renta)) %>%
  arrange(med.income) %>%
  mutate(city=factor(nomprov,levels=nomprov)) %>%
  ggplot(aes(x=city,y=med.income)) +
  geom_point(color="blue") +
  guides(color=FALSE) +
  xlab("City") +
  ylab("Median Income") +
  my_theme +
  theme(axis.text.x=element_blank(), axis.ticks = element_blank()) +
  geom_text(aes(x=city,y=med.income,label=city),angle=90,hjust=-.25) +
  theme(plot.background=element_rect(),
        panel.grid =element_blank(),
        axis.title =element_text(color="blue"),
        axis.text  =element_text(color="blue"),
        plot.title =element_text(color="blue")) +
  ylim(c(60000,180000)) +
  ggtitle("Income Distribution by City")

new.incomes <-data %>% select(nomprov) %>%
                 merge(data %>% group_by(nomprov) %>%
                    summarise(med.income=median(renta,na.rm=TRUE)),by="nomprov")
%>%
                 select(nomprov,med.income) %>%
                 arrange(nomprov)
data <- arrange(data,nomprov)
data$renta[is.na(data$renta)] <- new.incomes$med.income[is.na(data$renta)]

data$renta[is.na(data$renta)] <- median(data$renta,na.rm=TRUE)


test_data$renta <- as.numeric(test_data$renta)

new.incomes <-test_data %>% select(nomprov) %>%
                 merge(test_data %>% group_by(nomprov) %>%
                    summarise(med.income=median(renta,na.rm=TRUE)),by="nomprov")
%>%
                 select(nomprov,med.income) %>%
                 arrange(nomprov)
test_data <- arrange(test_data,nomprov)
test_data$renta[is.na(test_data$renta)] <-
new.incomes$med.income[is.na(test_data$renta)]

test_data$renta[is.na(test_data$renta)] <- median(test_data$renta,na.rm=TRUE)

```
```
```

```{r}
char.cols <- names(data)[sapply(data,is.character)]
for (name in char.cols){
  print(sprintf("Unique values for %s:", name))
  print(unique(data[[name]]))
  cat('\n')
}



char.cols <- names(test_data)[sapply(test_data,is.character)]
for (name in char.cols){
  print(sprintf("Unique values for %s:", name))
  print(unique(test_data[[name]]))
  cat('\n')
  }
```

Converting all character variables features into numeric variables

```{r}
data[,prod_cols] <- lapply(data[,prod_cols],function(x)as.integer(round(x)))

```

```{r}
unique_countries <- length(unique(data$pais_residencia))
top_10_countries <- data %>%
  group_by(pais_residencia) %>%
  summarise(count_by_countries=n())  %>%
  select(pais_residencia,count_by_countries) %>% arrange(-count_by_countries) %>%
head(10)
#kable(top_10_countries)
data$pais_residencia <- as.character(data$pais_residencia)
data$countries <- ifelse(data$pais_residencia %in% c('ES'), data$pais_residencia,'Other')

unique_countries <- length(unique(test_data$pais_residencia))
top_10_countries <- test_data %>%
  group_by(pais_residencia) %>%
  summarise(count_by_countries=n())  %>%
  select(pais_residencia,count_by_countries) %>% arrange(-count_by_countries) %>%
head(10)
test_data$pais_residencia <- as.character(test_data$pais_residencia)
```

```
unique_channels <- length(unique(data$canal_entrada))
top_10_channels <- data %>%
  group_by(canal_entrada) %>%
  summarise(count_by_channels=n())  %>%
  select(canal_entrada,count_by_channels) %>% arrange(-count_by_channels) %>%
head(10)
#kable(top_10_channels)
data$canal_entrada <- as.character(data$canal_entrada)
data$channel <- ifelse(data$canal_entrada %in% c('KHE', 'KAT', 'KFC','UNKNOWN'),
data$canal_entrada,'Other')

unique_channels <- length(unique(test_data$canal_entrada))
top_10_channels <- test_data %>%
  group_by(canal_entrada) %>%
  summarise(count_by_channels=n())  %>%
  select(canal_entrada,count_by_channels) %>% arrange(-count_by_channels) %>%
head(10)
test_data$canal_entrada <- as.character(test_data$canal_entrada)
test_data$channel <- ifelse(test_data$canal_entrada %in% c('KHE', 'KAT',
'KFC','UNKNOWN'), test_data$canal_entrada,'Other')

unique_provinces <- length(unique(data$nomprov))
top_10_provinces <- data %>%
  group_by(nomprov) %>%
  summarise(count_by_provinces=n())  %>%
  select(nomprov,count_by_provinces) %>% arrange(-count_by_provinces) %>%
head(10)
#kable(top_10_provinces)
data$nomprov <- as.character(data$nomprov)
data$provinces <- ifelse(data$nomprov %in% c('MADRID', 'BARCELONA',
'VALENCIA'), data$nomprov,'Other')

unique_provinces <- length(unique(test_data$nomprov))
top_10_provinces <- test_data %>%
  group_by(nomprov) %>%
  summarise(count_by_provinces=n())  %>%
  select(nomprov,count_by_provinces) %>% arrange(-count_by_provinces) %>%
head(10)
test_data$nomprov <- as.character(test_data$nomprov)
test_data$provinces <- ifelse(test_data$nomprov %in% c('MADRID', 'BARCELONA',
'VALENCIA'), test_data$nomprov,'Other')
```

#Exploratory Data Analysis:

#1. Analysis of Customer Initiation into the Bank by Month:-

```{r}
data <- as.data.table(data)
ggplot(data[year_alta>2009,.N, by =.(month_alta,year_alta)],aes(x = 
month_alta,y=N,fill=month_alta,))+
  geom_bar(stat="identity")+ggtitle("Number of customers that became 'first holder' by 
month and year")+
  facet_wrap(~year_alta) +scale_fill_manual(values=c("pink", "light 
blue","orange","violet","#DFFF00","#FFBF00","dark 
green","#DE3163","#9FE2BF","dark blue","#6495ED","#CCCCFF"))
```

#2. Analysis of Customer Age and Segmentation

```{r}
age_segmento <- ggplot(data, aes(x=age)) + 
  geom_bar(
    aes(fill=segmento )
  ) + 
  labs(title="Customer Age and Segmentation") +
  labs(x="Age", y="Number of Customers") +
  scale_fill_discrete(name = "Segmentation",
                    labels = c("VIP", "Individuals","College Graduated","Unnoted"))+
  my_theme + scale_fill_manual(values=c("#999999", "#E69F00", 
"#56B4E9","#FFC0CB"))
age_segmento

```

#3. Analysis on Channel of Entry and Household Income

```{r}
income_segment <- ggplot(data, aes(renta)) + 
  geom_histogram(breaks=seq(1203, 155500*3, by = 2000), 
               aes(fill=segmento)) + 
  labs(title="Histogram for Gross income of the household by Segment") +
  labs(x="Gross income of the household", y="Count")  +
  my_theme + scale_fill_manual(values=c("dark green", "blue", "yellow","red"))
income_segment
```

#4.Analysis on Customer Age and Channel.

```{r}
age_channel <- ggplot(data, aes(x=age))+  geom_bar(aes(fill=channel))+ xlab("Age") +
ylab("Number of Customers")+ ggtitle("Customer Age and Channel") + my_theme +
scale_x_discrete(limit = c(0,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100))
age_channel + scale_fill_manual(values=c(" green", "blue", "maroon","purple","red"))
```

#5. Analysis on Gross Income and channel.

```{r}
income_channel <- ggplot(data, aes(renta)) + geom_histogram(breaks=seq(1203,
155500*3, by = 2000),
                #col="red",
                aes(fill=channel)) +
  labs(title="Histogram for Gross income of the household by Channel") +
  labs(x="Gross income of the household", y="Count") +
  my_theme
income_channel
```

#6. Analysis of the Popularity of Products

```{r}
product_popularity_plot <- data %>% select(ind_ahor_fin_ult1:ind_recibo_ult1) %>%
summarise_each(funs(sum)) %>% gather(product, frequency,
ind_ahor_fin_ult1:ind_recibo_ult1) %>% ggplot(aes(x = reorder(product,frequency), y =
frequency)) + geom_bar(stat="identity", position="dodge", fill="green") + labs(y =
"Product Ownership Frequency", x = "Product") + my_theme + geom_text(aes(label =
frequency), position=position_dodge(width=1.5))+ coord_flip()
product_popularity_plot
```

#7. Analysis on Product Popularity by Segments

```{r}
product_popularity_per_segment <- data %>% group_by(segmento) %>%
select(segmento:ind_recibo_ult1) %>% summarise_each(funs(sum)) %>%
gather(product, frequency, ind_ahor_fin_ult1:ind_recibo_ult1) %>%  ggplot(aes(x =
product, y = frequency)) + geom_bar(stat="identity", position="dodge",
aes(fill=segmento)) + labs(y = "Product Ownership Frequency", x = "Product") +
my_theme + facet_wrap(~segmento) + theme(strip.text.x = element_text(size = 8, colour
= "black")) + coord_flip()
```

```
product_popularity_per_segment
```

```{r}
segments_per_product <- data %>% group_by(segmento) %>%
select(segmento:ind_recibo_ult1) %>% summarise_each(funs(sum)) %>%
gather(product, frequency, ind_ahor_fin_ult1:ind_recibo_ult1) %>%  ggplot(aes(x =
segmento, y = frequency)) + geom_bar(stat="identity", position="dodge",
aes(fill=segmento)) + labs(y = "Product Ownership Frequency", x = "Segment") +
my_theme + facet_wrap(~product) + theme(strip.text.x = element_text(size = 8, colour =
"black")) + coord_flip()
segments_per_product

```

#8. Analysis on Product popularity by Sex

```{r}
product_popularity_per_sex <- data[sexo!="UNKNOWN",] %>% group_by(sexo) %>%
select(sexo, ind_ahor_fin_ult1:ind_recibo_ult1) %>% summarise_each(funs(sum)) %>%
gather(product, frequency, ind_ahor_fin_ult1:ind_recibo_ult1) %>%  ggplot(aes(x =
product, y = frequency)) + geom_bar(stat="identity", position="dodge", aes(fill=sexo)) +
labs(y = "Product Ownership Frequency", x = "Product") + my_theme +
facet_wrap(~sexo) + theme(strip.text.x = element_text(size = 8, colour = "black")) +
coord_flip()
product_popularity_per_sex
```

```{r}
sex_per_product <- data[sexo!="UNKNOWN",] %>% group_by(sexo) %>% select(sexo,
ind_ahor_fin_ult1:ind_recibo_ult1) %>% summarise_each(funs(sum)) %>%
gather(product, frequency, ind_ahor_fin_ult1:ind_recibo_ult1) %>%  ggplot(aes(x =
sexo, y = frequency)) + geom_bar(stat="identity", position="dodge", aes(fill=sexo)) +
labs(y = "Product Ownership Frequency", x = "Sex") + my_theme +
facet_wrap(~product) + theme(strip.text.x = element_text(size = 8, colour = "black")) +
coord_flip()
sex_per_product
```

#Association Rule Mining and Market Basket Analysis of Products:-

```{r}
mb_data <- data %>% select(ncodpers,ind_ahor_fin_ult1:ind_recibo_ult1) %>%
gather(product,ownership,ind_ahor_fin_ult1:ind_recibo_ult1)
mb_data <- mb_data[mb_data$ownership==1,]
mb_data$ownership = NULL
```

````
mb_data_transactions <- split(mb_data$product, mb_data$ncodpers)
lapply(mb_data_transactions, write, "market_basket_data.txt", append=TRUE,
ncolumns=25)
mb_data_transactions <- read.transactions("market_basket_data.txt", sep=" ")

itemFrequencyPlot(mb_data_transactions, topN=10, type="absolute", main="Item
Frequency")
frequentProducts <- eclat (mb_data_transactions, parameter = list(supp = 0.05, maxlen =
15))

rules <- apriori (mb_data_transactions, parameter = list(supp = 0.05, conf = 0.8))
rules_conf <- sort (rules, by="confidence", decreasing=TRUE)
rules_lift <- sort(rules, by="lift", decreasing=TRUE)

#Remove Redundant Rules
subsetRules <- which(colSums(is.subset(rules, rules)) > 1) # get subset rules in vector
rules <- rules[-subsetRules] # remove subset rules
```

#Frequent Items –

```{r}
kable(inspect(frequentProducts))

```

#Association Rules sorted by Confidence
```{r}
kable(inspect(rules_conf))

```


#Association Rules sorted by Lift
```{r}
kable(inspect(rules_lift))

```

#Modeling and Performance Evaluation
```{r}
labels = colnames(data)[17:40]
data <- as.data.frame(data)
data[,17:40] <- apply(as.matrix(data[,17:40]),2, function(x) as.logical(x));

# #data <- subset(data, select=(-c(ind_aval_fin_ult1)))
````

~~~~
# labels <- labels[!c("ind_aval_fin_ult1") %in% labels]
# data<- data %>% drop_na

data$antiguedad <- as.integer(data$antiguedad)
data$month_alta <- factor(data$month_alta, ordered = FALSE)

model_data <- data[,c(-1,-3,-6,-12,-13)]

model_data$ind_empleado <- as.factor(model_data$ind_empleado)
model_data$indresi <- as.factor(model_data$indresi)
model_data$provinces <- as.factor(model_data$provinces)
model_data$channel <- as.factor(model_data$channel)
model_data$countries <- as.factor(model_data$countries)

mysample <- sample(1:nrow(model_data),0.7*nrow(model_data))
train_data <- model_data[mysample,]
test_data <- model_data[-mysample,]

train_data_rfsrc <- train_data
train_data_rfsrc$month_alta <- factor(train_data_rfsrc$month_alta, ordered = FALSE)
train_data_rfsrc_sample <- sample(1:nrow(train_data_rfsrc),0.3*nrow(train_data_rfsrc))
train_data_rfsrc <- train_data_rfsrc[train_data_rfsrc_sample,]
test_data_rfsrc <- train_data_rfsrc[-train_data_rfsrc_sample,]
test_data_rfsrc <-
test_data_rfsrc[sample(1:nrow(test_data_rfsrc),0.5*nrow(test_data_rfsrc)),]
# train_data_rfsrc$antiguedad <- as.integer(train_data_rfsrc$antiguedad)
# train_data$antiguedad <- as.integer(train_data$antiguedad)
products_task_train_rfsrc = makeMultilabelTask(id = "products_rfsrc", data =
train_data_rfsrc, target = labels)
train_data <- na.omit(train_data)
products_task_train = makeMultilabelTask(id = "products", data = train_data, target =
labels)
```

# Binary Relevance Problem Transformation Method

```{r}
library(mlr)
#Constructing the Learner
learn_br_prob = makeLearner("classif.rpart", predict.type = "prob")
learn_br_prob = makeMultilabelBinaryRelevanceWrapper(learn_br_prob)

#Model Training
br_model = mlr::train(learn_br_prob, products_task_train)

#Model Prediction
~~~~

```
br_model_pred = predict(br_model, task = products_task_train)

#removed na because below predict fn was giving error
test_data <- test_data %>% drop_na()

br_model_pred_test = predict(br_model, newdata=test_data)

#Model Performance

br_model_perf <- performance(br_model_pred, measures = list(multilabel.subset01,
multilabel.hamloss, multilabel.acc, multilabel.f1, timepredict))
br_model_perf_test <- performance(br_model_pred_test, measures =
list(multilabel.subset01, multilabel.hamloss, multilabel.acc, multilabel.f1, timepredict))

getMultilabelBinaryPerformances(br_model_pred_test, measures = list(mlr::acc, mlr::fpr,
mlr::tpr, mlr::auc))

library(pROC)
#roc for ind_ctju_fin_ult1

pp <- br_model_pred$data$truth.ind_ctju_fin_ult1
qq <- br_model_pred$data$prob.ind_ctju_fin_ult1
roc1 <- roc(pp, qq, plot = TRUE, print.auc = TRUE,legacy.axes = TRUE, legend=TRUE)

#roc for ind_ctju_fin_ult1

pp <- br_model_pred$data$truth.ind_cco_fin_ult1
qq <- br_model_pred$data$prob.ind_cco_fin_ult1
roc2<- roc(pp, qq, plot = TRUE, print.auc = TRUE,legacy.axes = TRUE, legend =
TRUE)

#roc for ind_ctju_fin_ult1

pp <- br_model_pred$data$truth.ind_ctop_fin_ult1
qq <- br_model_pred$data$prob.ind_ctop_fin_ult1
roc3 <- roc(pp, qq, plot = TRUE, print.auc = TRUE,legacy.axes = TRUE, legend =
TRUE)

#roc for ind_dela_fin_ult1

pp <- br_model_pred$data$truth.ind_dela_fin_ult1
qq <- br_model_pred$data$prob.ind_dela_fin_ult1
roc4 <- roc(pp, qq, plot = TRUE, print.auc = TRUE,legacy.axes = TRUE, legend =
TRUE)

#roc for ind_ecue_fin_ult1
```

pp <- br_model_pred$data$truth.ind_ecue_fin_ult1
qq <- br_model_pred$data$prob.ind_ecue_fin_ult1
roc5 <- roc(pp, qq, plot = TRUE, print.auc = TRUE,legacy.axes = TRUE, legend = TRUE)

plot(roc1, col = 1, lty = 2, main = "ROC Cure for Binary Relevance Model ", legacy.axes = TRUE)
plot(roc2, col = 4, lty = 3, add = TRUE, legacy.axes = TRUE)
plot(roc3, col = 7, lty = 4, add = TRUE, legacy.axes = TRUE)
plot(roc4, col = 11, lty = 5, add = TRUE, legacy.axes = TRUE)

plot(roc5, col = 15, lty = 6, add = TRUE, legacy.axes = TRUE)

legend("bottomright", legend=c("ind_ctju_fin_ult1", "ind_ctju_fin_ult1","ind_dela_fin_ult1","ind_dela_fin_ult1", "ind_ecue_fin_ult1"), col = c("red","blue","green","orange","pink"), lty=1:2, cex=0.8, text.font=4, bg='lightblue')

```

# logistic

```{r}
library(e1071)

logistic.learner <- makeLearner("classif.logreg",predict.type = "response")
logistic.learner <- makeMultilabelBinaryRelevanceWrapper(logistic.learner)
#products_task_train$env$data <- products_task_train$env$data %>% drop_na()

# train model
logistic.model <- mlr::train(logistic.learner, products_task_train)

# predict on test data
logistic.model.pred <- predict(logistic.model, newdata=test_data)

logistic.model_perf_test <- performance(logistic.model.pred, measures = list(multilabel.subset01, multilabel.hamloss, multilabel.acc, multilabel.f1, timepredict))

# cross validation (10 fold logistic regression) (cv) accuracy
cv.logistic <- crossval(learner = logistic.learner,task = products_task_train,iters = 10, show.info = T)

```
```

```
# Random Forests Algorithm Adapatation Method:-

```{r}
#Constructing the Learner

lrn.rfsrc = makeLearner("multilabel.randomForestSRC")

#Model Training
rfsrc_model = train(lrn.rfsrc,products_task_train_rfsrc)
rfsrc_model

#Model Prediction

rfsrc_model_pred = predict(rfsrc_model, task=products_task_train_rfsrc)
# removed na as predict fn below was giving error
test_data_rfsrc <- test_data_rfsrc %>% drop_na()

rfsrc_model_pred_test = predict(rfsrc_model, newdata=test_data_rfsrc)

#Model Performance

rfsrc_model_perf <- performance(rfsrc_model_pred, measures = list(multilabel.subset01,
multilabel.hamloss, multilabel.acc, multilabel.f1, timepredict))
rfsrc_model_perf_test <- performance(rfsrc_model_pred_test, measures =
list(multilabel.subset01, multilabel.hamloss, multilabel.acc, multilabel.f1, timepredict))


#cross validation (10 fold random forest) (cv) accuracy

lrn.rfsrc = makeLearner("multilabel.randomForestSRC")
cv.logistic <- crossval(learner = lrn.rfsrc,task = products_task_train_rfsrc,iters =
10,show.info = F)

# Feature importance

# im_feat <- generateFilterValuesData(products_task_train_rfsrc, method =
c("information.gain","chi.squared"))
# plotFilterValues(im_feat,n.show = 20)
```

#Neural Networks Algorithm Adaptation Method


```{r}
model_data_nn <- model_data
```
```

```
model_data_nn[,12:35] <- apply(model_data[,12:35], 2, function(x) as.numeric(x));
model_data_nn$segmento <- as.character(model_data_nn$segmento)
model_data_nn[model_data_nn$segmento == "02 - PARTICULARES", "segmento"] <-
"PARTICULARES"
model_data_nn[model_data_nn$segmento == "03 - UNIVERSITARIO", "segmento"] <-
"UNIVERSITARIO"
model_data_nn[model_data_nn$segmento == "01 - TOP", "segmento"] <- "TOP"
col_names <- names(model_data_nn)
formula <- as.formula(paste(paste(labels, collapse="+"),"~",
paste(col_names[!col_names %in% labels], collapse = " + ")))

m <- model.matrix(~ind_ahor_fin_ult1 + ind_aval_fin_ult1 + ind_cco_fin_ult1 +
ind_cder_fin_ult1 +
                ind_cno_fin_ult1 + ind_ctju_fin_ult1 + ind_ctma_fin_ult1 +
                ind_ctop_fin_ult1 + ind_ctpp_fin_ult1 + ind_deco_fin_ult1 +
                ind_deme_fin_ult1 + ind_dela_fin_ult1 + ind_ecue_fin_ult1 +
                ind_fond_fin_ult1 + ind_hip_fin_ult1 + ind_plan_fin_ult1 +
                ind_pres_fin_ult1 + ind_reca_fin_ult1 + ind_tjcr_fin_ult1 +
                ind_valo_fin_ult1 + ind_viv_fin_ult1 + ind_nomina_ult1 +
                ind_nom_pens_ult1 + ind_recibo_ult1 + ind_empleado + sexo +
                age + ind_nuevo + antiguedad + indrel + tiprel_1mes + indresi +
                ind_actividad_cliente + renta + segmento + provinces + countries +
                channel, data=model_data_nn)
m <- as.data.frame(m)
m <- m[,c(-1)]
col_names <- names(m)
formula <- as.formula(paste(paste(labels, collapse="+"),"~",
paste(col_names[!col_names %in% labels], collapse = " + ")))

train_data_nn <- m[mysample,]
test_data_nn <- m[-mysample,]

train_data_nn[is.na(train_data_nn)] <- 0

neural_net <- neuralnet(formula,
            data = train_data_nn,
            act.fct = "logistic",
            linear.output = FALSE,
            lifesign = "minimal")
plot(neural_net)
#Compute Predictions
neural_net_predict <- compute(neural_net, train_data_nn[, c(25:50)])
neural_net_predict_test <- compute(neural_net, test_data_nn[,25:50])


# Extract Results
```

```
neural_net_predictions <- neural_net_predict$net.result
neural_net_predictions <- apply(neural_net_predictions,2,function(x) round(x))
head(neural_net_predictions)

neural_net_predictions_test <- neural_net_predict_test$net.result
neural_net_predictions_test <- apply(neural_net_predictions_test,2,function(x) round(x))


# Accuracy (training set)
original_values <- train_data_nn[, 1:24]
row_match <- as.array(seq(from=0,to=0,length.out=dim(train_data_nn)[1]))
row_mismatch <- as.array(seq(from=0,to=0,length.out=dim(train_data_nn)[1]))

for(i in 1:dim(neural_net_predictions)[1])
  {for(j in 1:dim(neural_net_predictions)[2])
  {
   if(neural_net_predictions[i,j]==original_values[i,j]){ row_match[i]=row_match[i]+1}
   else {row_mismatch[i]=row_mismatch[i]+1}
  }
}
nnet_perf_train <- mean((row_match-row_mismatch)/24)



# Accuracy (test set)
original_values <- test_data_nn[, 1:24]
row_match <- as.array(seq(from=0,to=0,length.out=dim(test_data_nn)[1]))
row_mismatch <- as.array(seq(from=0,to=0,length.out=dim(test_data_nn)[1]))

for(i in 1:dim(neural_net_predictions_test)[1])
  {for(j in 1:dim(neural_net_predictions_test)[2])
  {
   if(neural_net_predictions_test[i,j]==original_values[i,j]){
row_match[i]=row_match[i]+1}
   else {row_mismatch[i]=row_mismatch[i]+1}
  }
}

nnet_perf_test <- mean((row_match-row_mismatch)/24)
nnet_perf_test

```

```
```{r}
```
```

```
# Binary Relevance Performance

kable(br_model_perf)
kable(br_model_perf_test)

#Random Forests Performance

kable(rfsrc_model_perf)
kable(rfsrc_model_perf_test)

# Neural Net Accuracy

kable(nnet_perf_train)
kable(nnet_perf_test)


```

References:

1. **https://arxiv.org/pdf/1904.10551.pdf**
2. **https://www.kaggle.com/c/santander-product-recommendation/**
3. **https://medium.com/@agupta.rkl/market-basket-analysis-using-association-rule-mining-66b61c0d5f26**
4. **https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/**