Retail Sales Strategy: Data Mining

Team: Anish Gandhi

Chanchal Sarkar

Sanjay Bhansali

|  |  |
| --- | --- |
| **Table Of Contents** | **Page Number** |
| **Business Overview** | **3** |
| **Business Requirement** | **3** |
| **Data Description** | **4** |
| **Regression Analysis** | **5-10** |
| **Classification : CHAID Tree** | **11-13** |
| **Classification : C5 Tree** | **14-15** |
| **Classification : Random Forest** | **16-18** |
| **Association Analysis** | **19-22** |

Business Overview:

Company X is an MNC headquartered in USA is into retail distribution and shipment business for office supplies, technology equipment and furniture. In 2012 company X license was cancelled in Canada due to some legal issues and now Canada has reinstated their license to perform business operations again. Company X was operational from 2009-2012 in Canada and in order to launch the operations again with wider range of products in each category it’s planning to make some strategic decisions by applying data mining techniques on the historical sales data the company has from 2009-2012. Company X is looking for some better classification and association rules which can help them to increase their business revenue and profit as well by offering wide variety of products bundled together based on the strong association rules.

Business Requirement:

Company X wants the data science team to apply data mining techniques and help the business to answer the following questions:

1 > Top KPI’s (Key Performance Indicator) driving profit

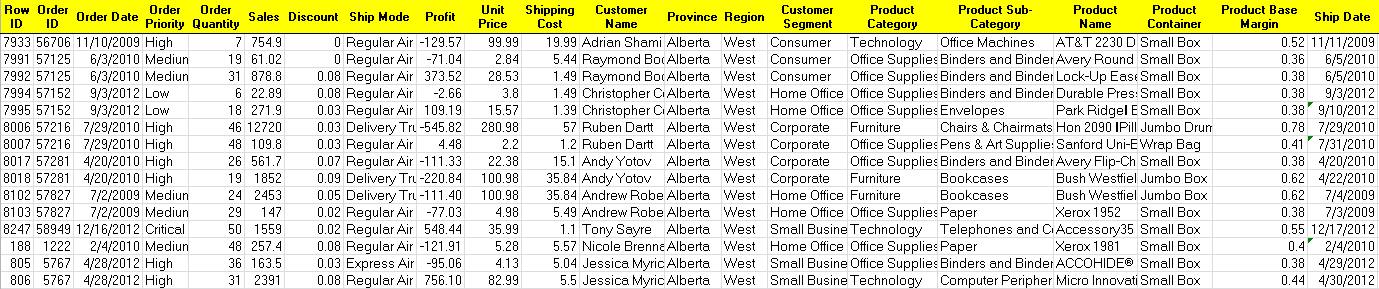
2 > Provide classification rules to classify whether the transaction is profitable or not

3 > Provide strong association rules on the product sub-categories which can be bundled together as an offer

Data Description:

Company X has about 8400 transactions spanning across 8 regions in Canada where different products are being shipped in different mode and in different sizes of a container .The data also includes information about the discount applied on each transaction, shipping cost, unit price, product base margin, overall sales. Initial data layout has about 21 variables to perform sales analysis.

Following is the sales transactional data snapshot which contains both quantitative and qualitative variables:



Regression Analysis:

We have used multivariate linear regression technique to identify the KPI’s driving profit.

Following is our strategy for regression:

1 > All the fields which has significance value (P) > 0.01 will be rejected

2 > There should be no violation of variance, normality,linearity,independence

3 > Variable transformation if required to achieve constant variance

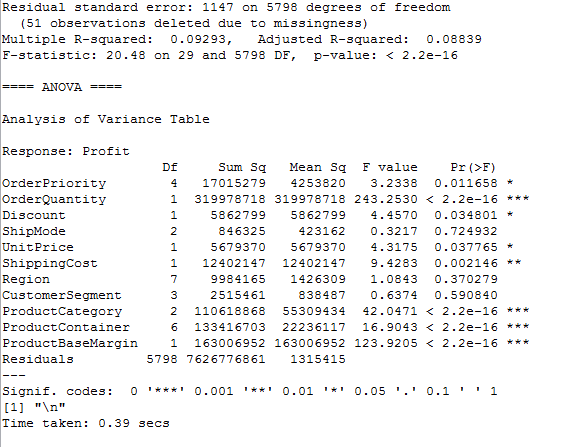
4 > Partition data as follow Training – 70 %, Test - 15 %, Validation - 15 %

5 > Out of 21 fields 8 fields are eliminated based on the business knowledge

**order id, row id, customer name,province,product name,product sub-category,order date,ship date**

6>Profit field which is present in each transaction set as the target variable

We ran the full model with 11 input variables and 1 target variable, following is the output of linear regression:

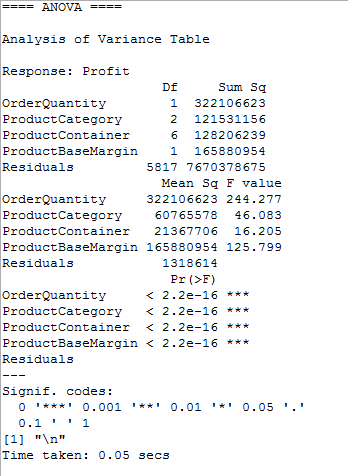


Result analysis of full regression model

As per full model following 7 fields are insignificant since p-value > 0.01.

Order Priority,Discount,Unit Price,Shipping Cost,Shipmode,Region,Customer Segment.

Now we have reduced model with 4 variables, below is the output of reduced model:



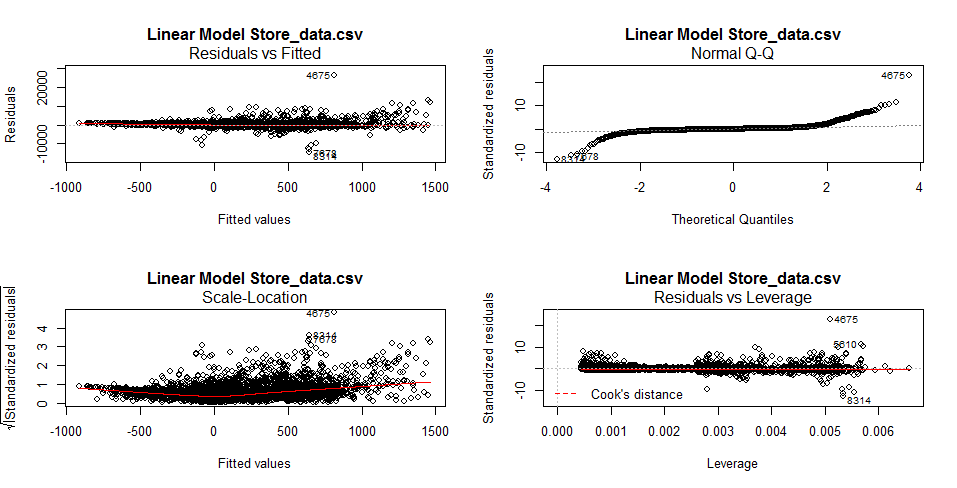
Result Analysis of the reduced model

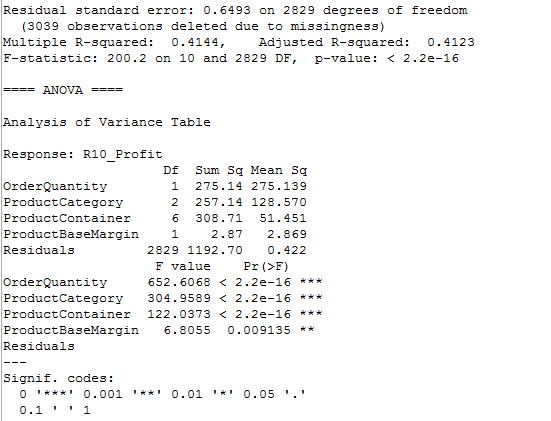
All fields are significant in the reduced model

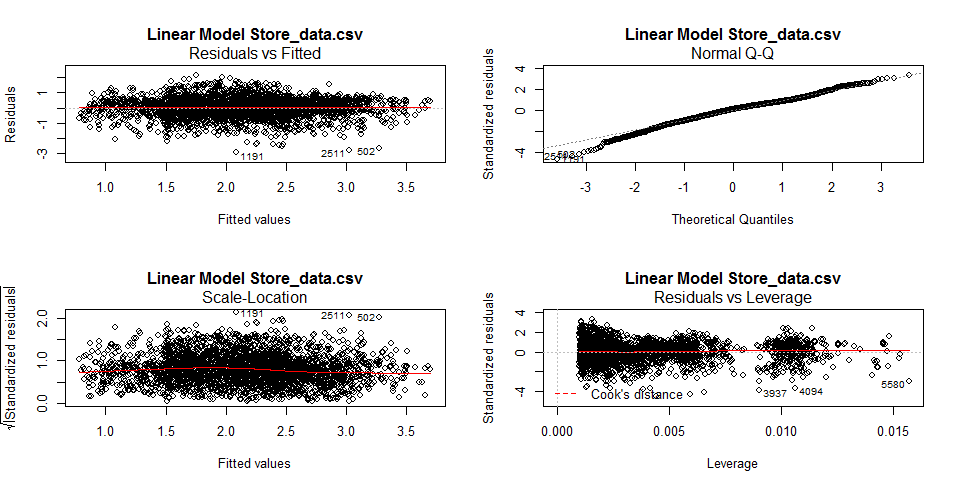
Below is the regression plot which indicates there is a violation of

constant variance and normality.

We applied log transformation on target variable to fit the data into model.







Result Analysis of the transformed model:

After transformation, there is no more violation of linear regression principles.

Top KPI’s driving profit is:

Order Quantity, Product Container, Product Category & Product Base Margin.

CLASSIFICATION:

In addition to regression, we wanted to see the factors that drive profitability,

so we derived a new variable “Profitable” as follow:

1 if profit is greater than zero

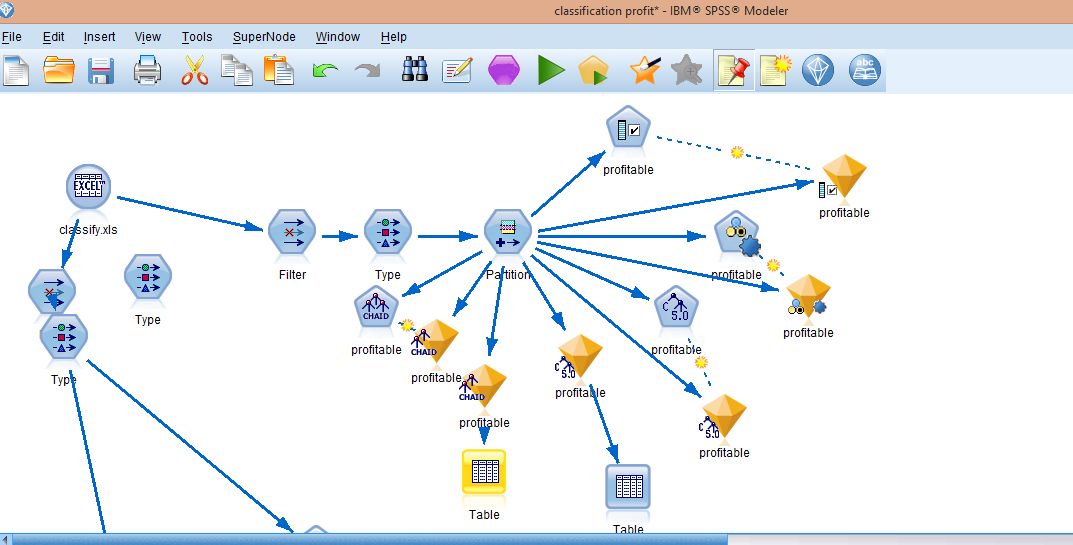
0 if profit is less than or equal to zero

We split the data into 70/15/15 for training/test/validation and used SPSS’s feature

selection methodology to come up with best trees to use, based on the importance factor,

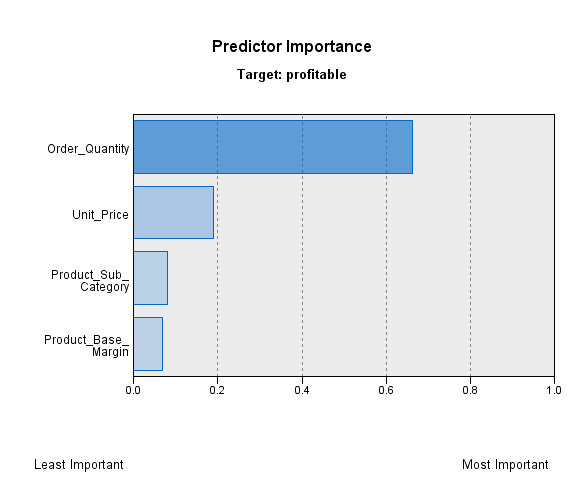
we decided to use CHAID and C5 tree.

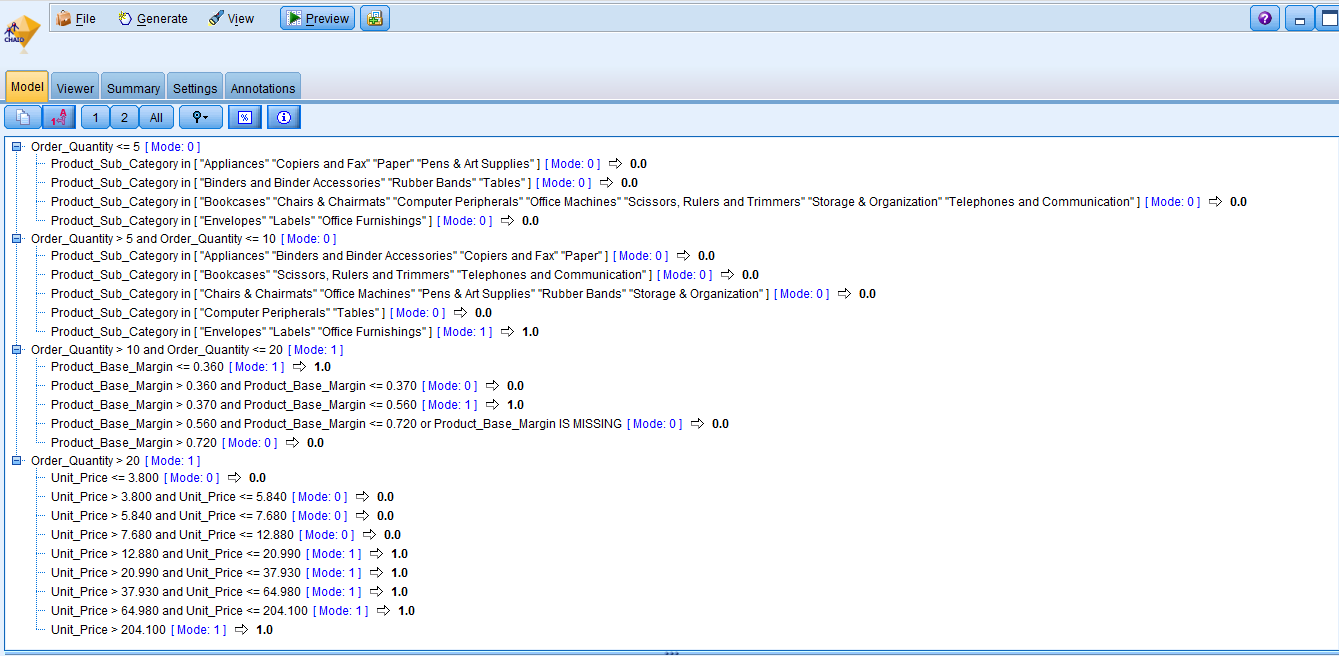
IBM SPSS Classification Model:

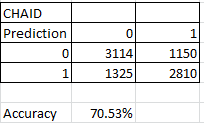


CHAID Tree:

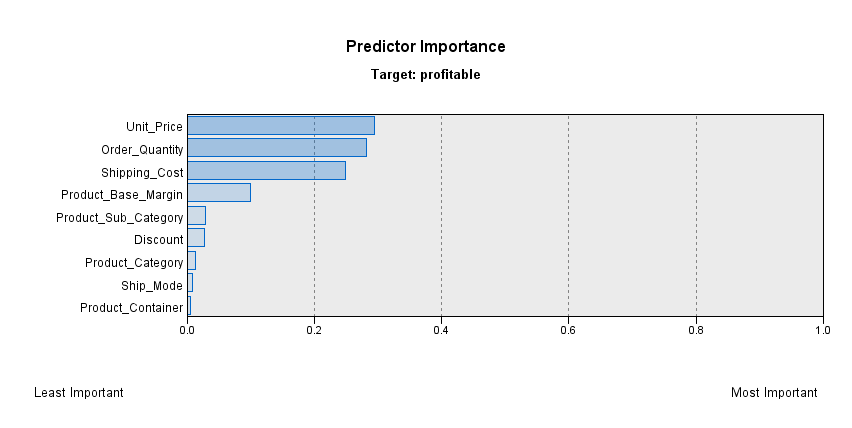
The **CHAID** Analysis (Chi Square Automatic Interaction Detection) is a form of analysis that determines how variables best combine to explain the outcome in a given dependent variable. CHAID analysis is especially useful for data expressing categorized values instead of continuous values. One of the outstanding advantages of CHAID analysis is that it can visualize the relationship between the target (dependent) variable and the related factors with a tree image.

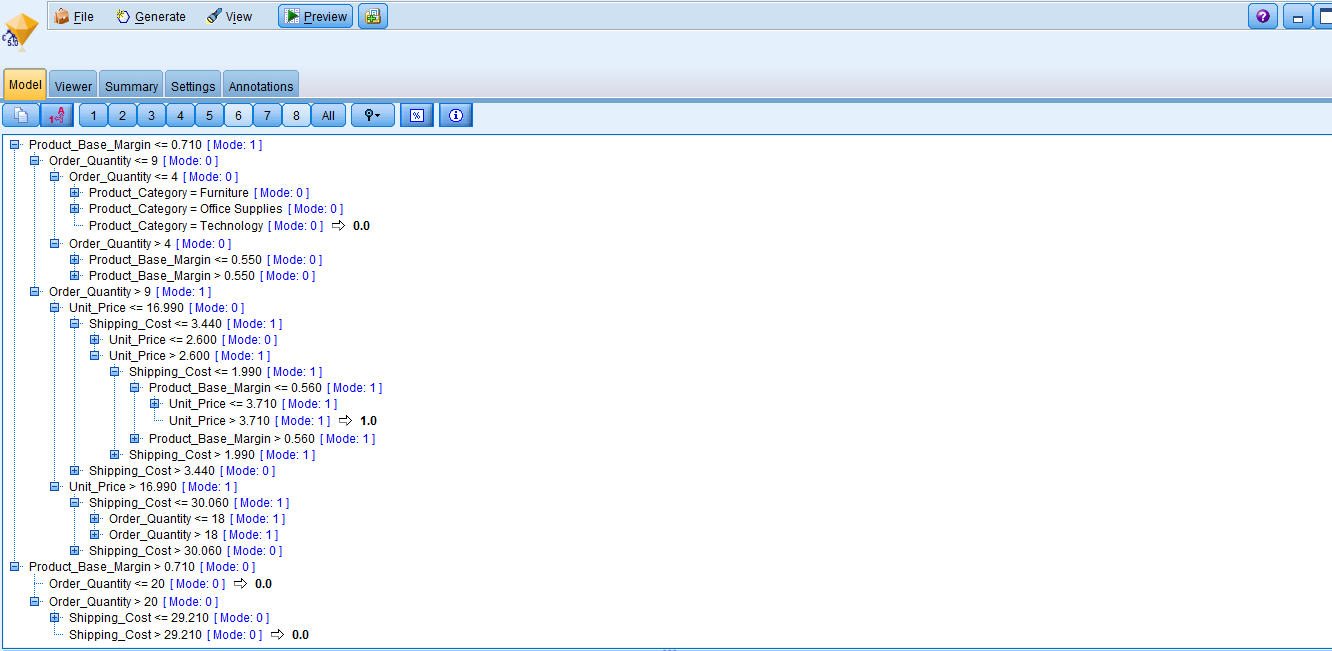


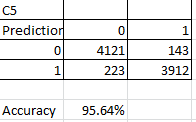




C5 Tree:







Overall C5 has much better accuracy then CHAID as per the validation results.

Ensemble Method: Random Forest

Ensemble methods are learning algorithms that construct a. set of classifiers and then classify new data points by taking a (weighted) vote of their predictions. The original ensemble method is Bayesian averaging, but more recent algorithms include error-correcting output coding, Bagging, and boosting.

**A unit or group of complementary parts that contribute to a single effect.Take a large collection of individually imperfect models, and their one-off mistakes are probably not going to be made by the rest of them. If we average the results of all these models, we can sometimes find a superior model from their combination than any of the individual parts. That’s how ensemble models work, they grow a lot of different models, and let their outcomes be averaged or voted across the group.**

**We are now well aware of the overfitting problems with decision trees. But if we grow a whole lot of them and have them vote on the outcome, we can get passed this limitation.**

**Random Forest models grow trees much deeper than the decision stumps above, in fact the default behavior is to grow each tree out as far as possible, like the overfitting tree we made in. But since the formulas for building a single decision tree are the same every time, some source of randomness is required to make these trees different from one another. Random Forests do this in two ways. The first trick is to use bagging, for bootstrap aggregating. Bagging takes a randomized sample of the rows in your training set, with replacement.**

**Since we do not have any missing values in the dataset, so we’re good to split the test and train sets back up and grow a Random Forest.**

**Install and load the package Random Forest:**

**install.packages('randomForest')**

**library(randomForest)**

**Now we’re ready to run our model.**

**sale <- read.csv("E:/Fall\_2015/Data Mining Class - CSE891\_002/project/classify.csv")**

**train <- sale[1:4199,]**

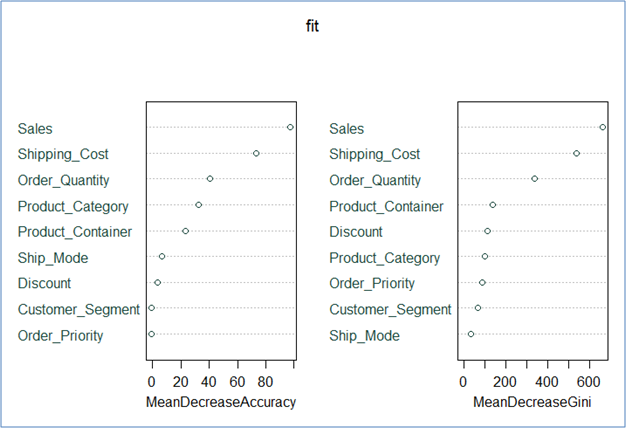
**test <- sale[4200:8399,]**

**fit <- randomForest(as.factor(profitable) ~ Order\_Priority + Order\_Quantity + Sales + Discount + Ship\_Mode + Shipping\_Cost + Customer\_Segment + Product\_Category + Product\_Container , data=train, importance=TRUE, ntree=100)**

**We force the model to predict our classification by changing our target variable to a factor with only two levels using as.factor (). The importance=TRUE argument allows us to inspect variable importance as we’ll see, and the ntree argument specifies how many trees we want to grow.**

**Since we only have a small dataset to play with, we can grow a large number of trees and not worry too much about their complexity, it will still run pretty fast.**

**So let’s look at what variables were important: varImpPlot(fit)**



**Bagging roughly excludes a third of our rows. Well Random Forests doesn’t just waste those “out-of-bag” (OOB) observations; it uses them to see how well each tree performs on unseen data. It’s almost like a bonus test set to determine your model’s performance on the fly.**

**There are two types of importance measures shown above. The accuracy one tests to see how worse the model performs without each variable, so a high decrease in accuracy would be expected for very predictive variables. The Gini one digs into the mathematics behind decision trees, but essentially measures how pure the nodes are at the end of the tree. Again it tests to see the result if each variable is taken out and a high score means the variable was important. Unsurprisingly, our sales variable was at the top for both measures.**

Accuracy: 88.79% 

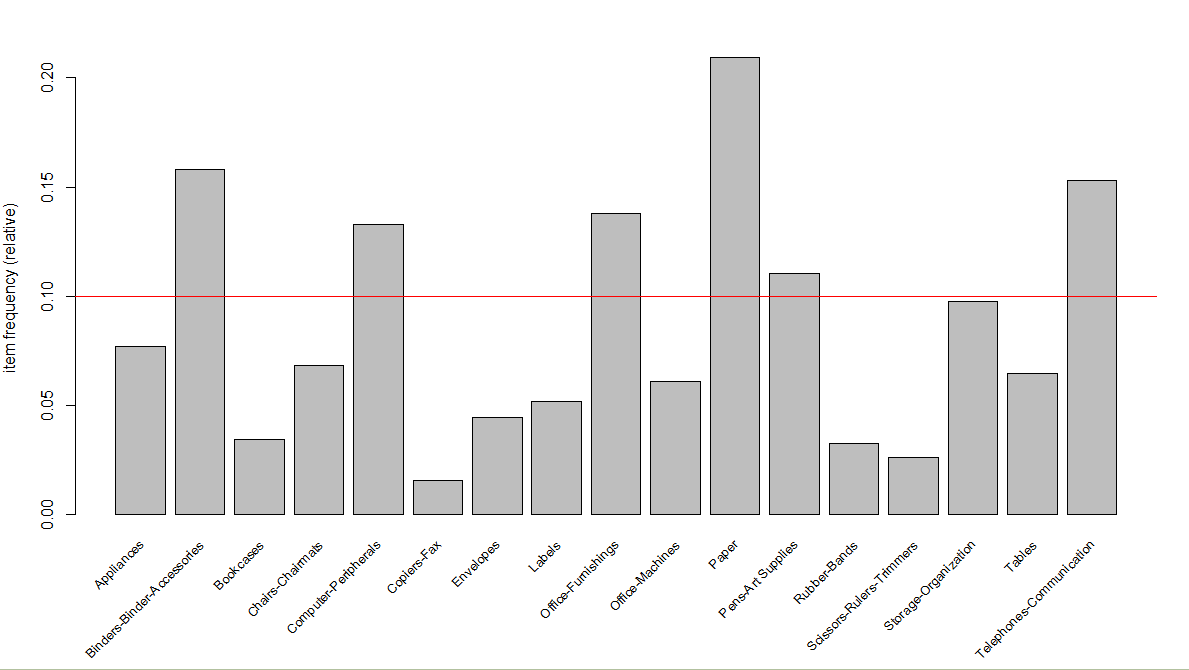
Association Analysis

Using Apriori algorithm, we have generated frequent itemset with size 1 with minsup > 10%. Overall we have 17 Product Sub -category, but only 6 Sub-Categories are meeting the minimum support threshold greater than 10%. Please find the list below:

|  |  |
| --- | --- |
| **Product Sub-Category** | **Support (%)** |
| Paper | 20.924 |
| Binders and Binder Accessories | 15.793 |
| Telephones and Communications | 15.284 |
| Office Furnishings | 13.774 |
| Computer Peripherals | 13.301 |
| Pens & Art Supplies | 11.044 |

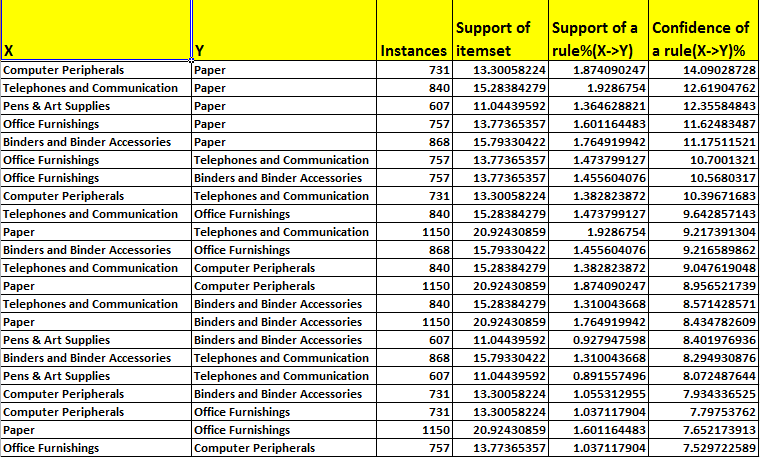
Relative Item Frequency plot indicates the sub-category support count with red line

Indicating the sub-category satisfying the criteria of minsup > 10 %

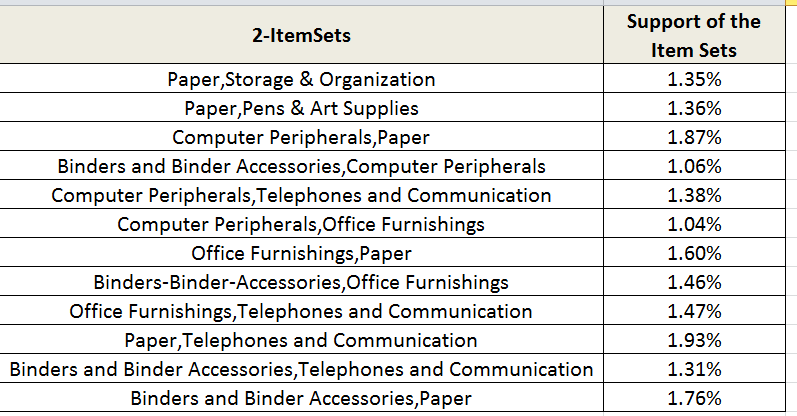


List of rules generated with the threshold:

rule support > 1 % and rule confidence > 7.5 %



Frequent Itemset – Size 2



Overall association analysis indicates that the rules generated do

not have high support and high confidence.

Also the frequent itemset are not appearing that frequent, so

Association analysis will not be much helpful.