

**Predictive Model for choosing correct shipping mode**

GROUP ASSIGNMENT

Build a predictive model to ascertain the correct shipping mode and choose the significant predictors wisely.

**CP G1 - PGP BABI Apr’19**

**Amit Sharma**

**Amitesh Bajpai**

**Sahil Sachdeva**

**Kushal Maheshwari**

**Neeraj Singh**

# Project Objective

The objective of the report is to build the best predictive model which can predict the correct shipping mode for a Supply Chain and Logistics company based on the given data “**09\_Inventory.xlsx**” in R, reflect upon the performance of the various models and find the best model and which variables are a significant predictor behind this decision. This exploration report will consist of the following:

* + Importing the dataset in R
  + Understanding the structure of dataset
  + Graphical exploration
  + Descriptive statistics
  + Oversampling/Undersampling with ROSE
  + One-Hot Encoding
  + Multinomial Logistic Regression
  + Support Vector Machine
  + Bagging
  + Boosting
  + Decision Tree
  + Random Forest
  + Gradient Boosting
  + Insights from the dataset

# Assumptions

The following assumptions are made for the inferential statistics:

1. Observations are independent
2. Samples are random
3. Measurements are accurate
4. For Naïve Bayes: The variables are independent and are equally important.
5. For KNN: We normalize the continuous variables

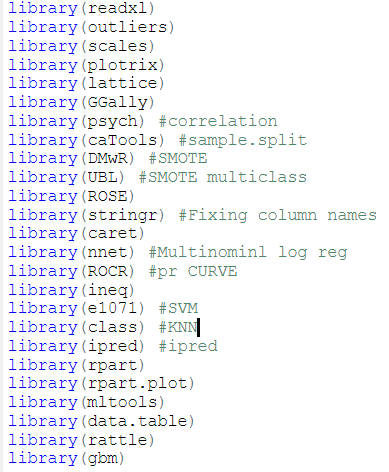
# Exploratory Data Analysis – Step by step approach

The various steps followed to analyze the case study is mentioned and explained below.

## Environment Set up and Data Import

### Install necessary Packages and Invoke Libraries

The lists of R packages used to analyze the data are listed below:



### Set up working Directory

Setting up the working directory will help to maintain all the files related to the project at one place in the system.

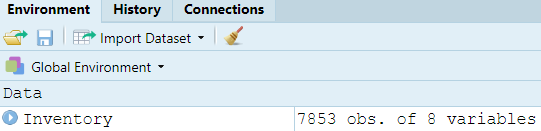
The working directory, we have setup in a local folder in the laptop.



Please refer Appendix A for Source Code.

### Import and Read the Dataset

The given dataset is in “.xlsx” format, so to import the data in R we use the “read\_xlsx” command. Data in file “09\_Inventory.xlsx” is stored in a variable called “**Inventory**”.



Please refer Appendix A for Source Code.

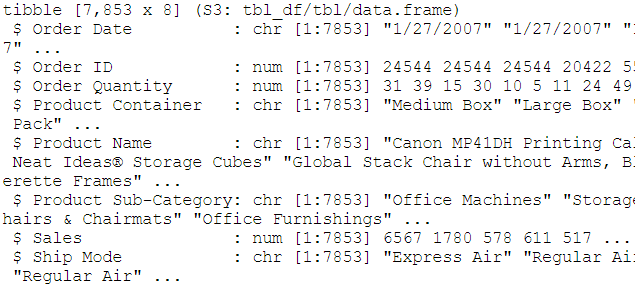
## Variable Identification

### Variable Identification – Inferences

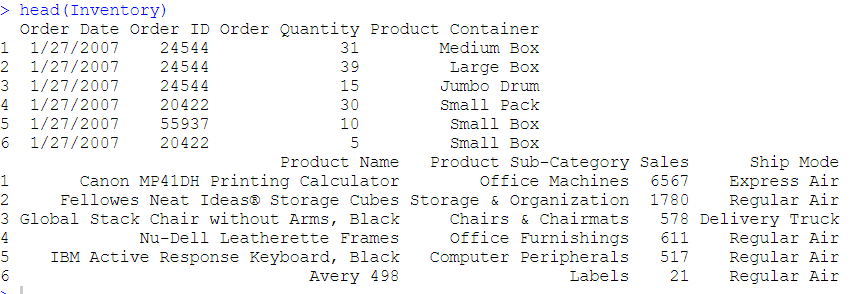
* **DIM**
  + Inventory data frame : There are 7853 rows and 8 columns



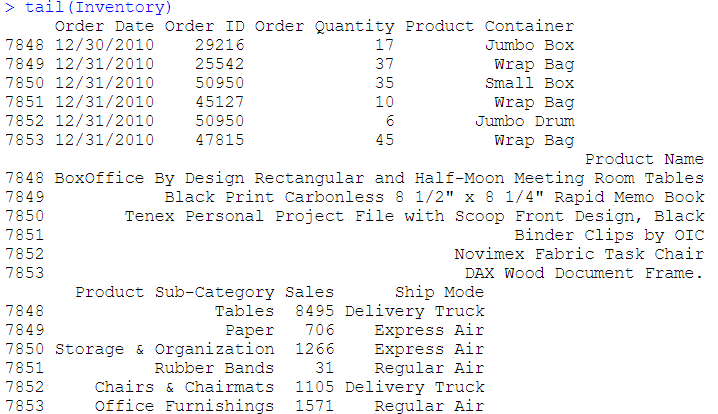
* **STR**
  + There are 8 variables in the dataset.



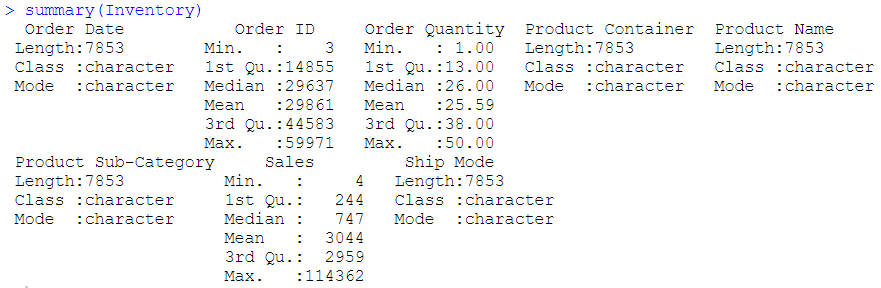
* **HEAD**
  + Inventory data frame: Verifying head records



* **TAIL**
  + Inventory data frame: Verifying tail records



* **SUMMARY**
  + Inventory data frame: The variables, namely, Ship Mode, Product Container and Product Sub-Category should be factor. So, we will convert them to factor. The continuous variables may have outliers.



Please refer Appendix A for Source Code.

## Missing Value Identification

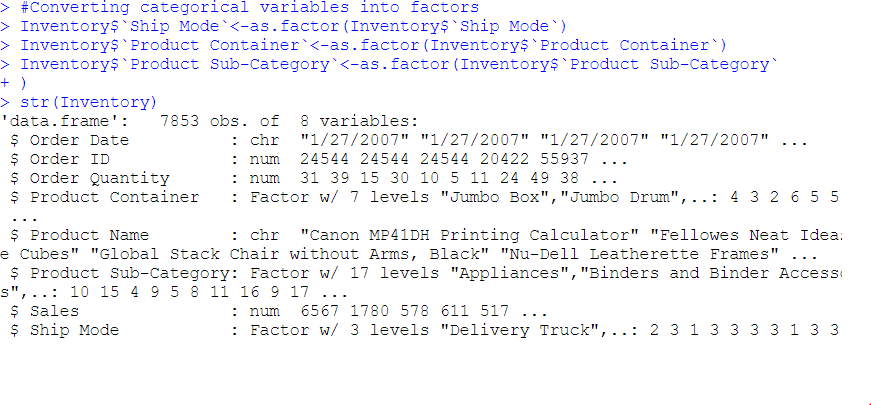
We use ‘is na’ function to check if there are any missing values. There are 0 missing values in the dataset.



Please refer Appendix A for Source Code.

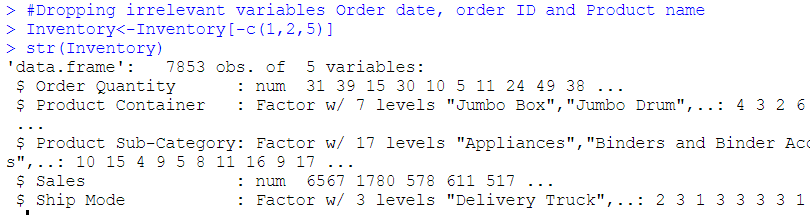
## Variable Transformation / Feature Creation/Deletion

In Summary of data we have seen that the "Ship Mode", "Product Container" and "Product Sub-Category" variables should be factor. So, we use ‘ as.factor ‘ to convert them.



Please refer Appendix A for Source Code.

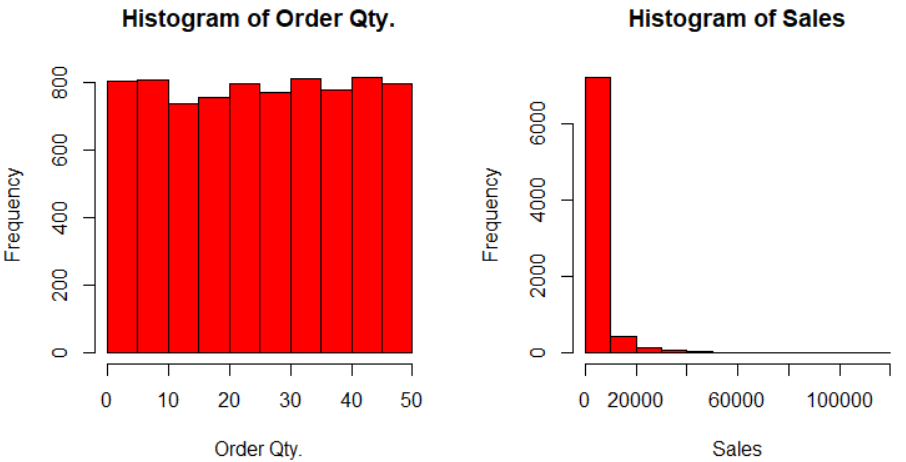
We see that Order date, Order ID and Product Name are irrelevant for prediction, so we take a decision to drop them completely.

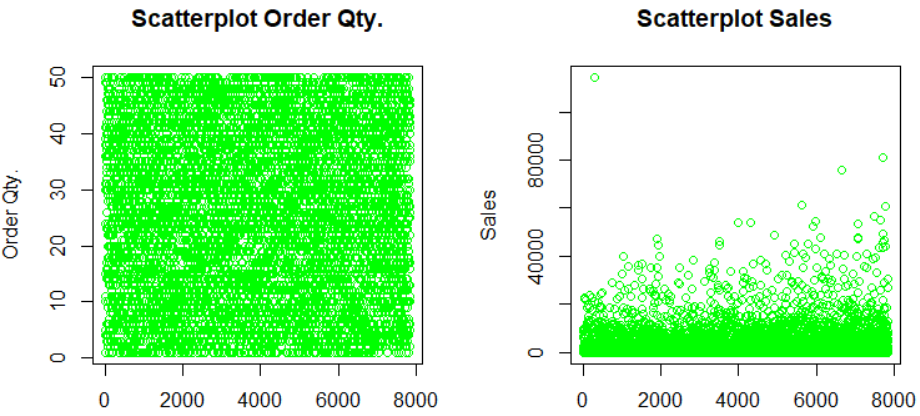


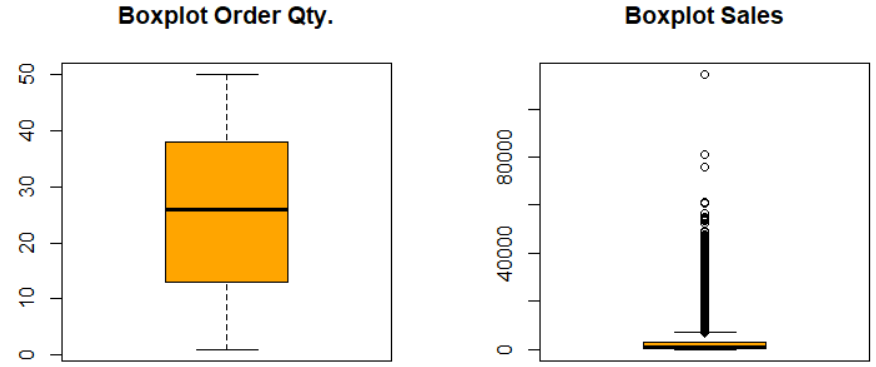
## Univariate Analysis

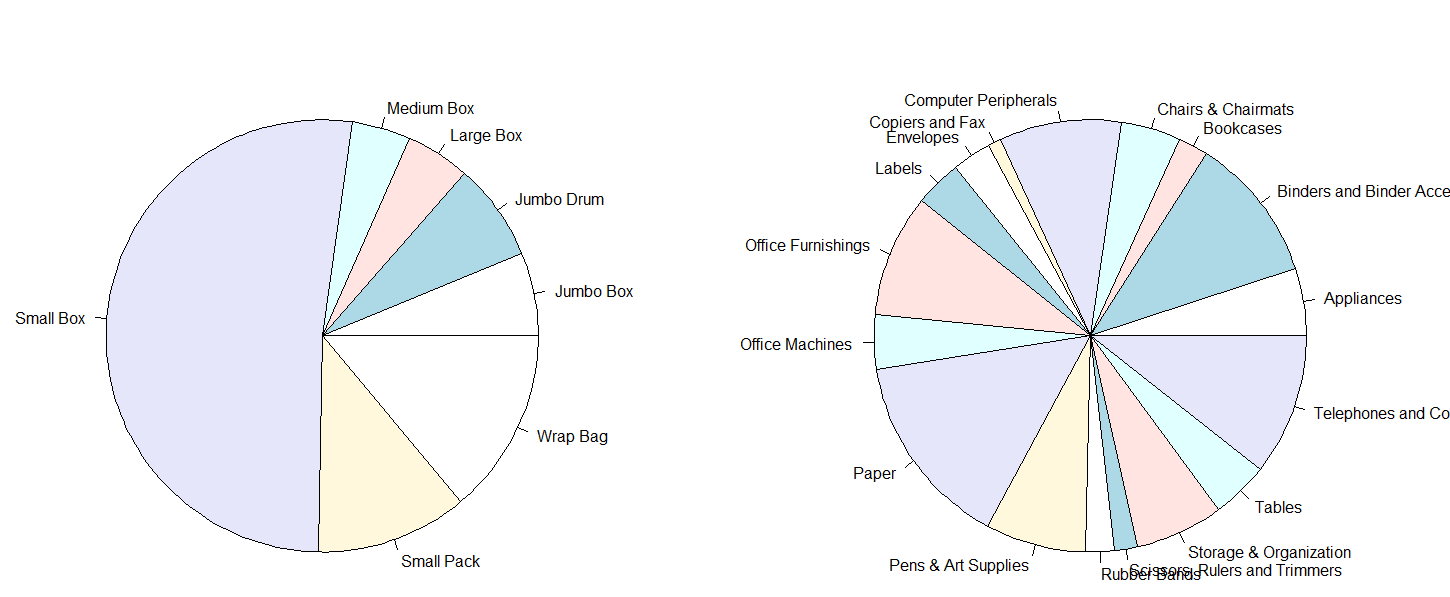
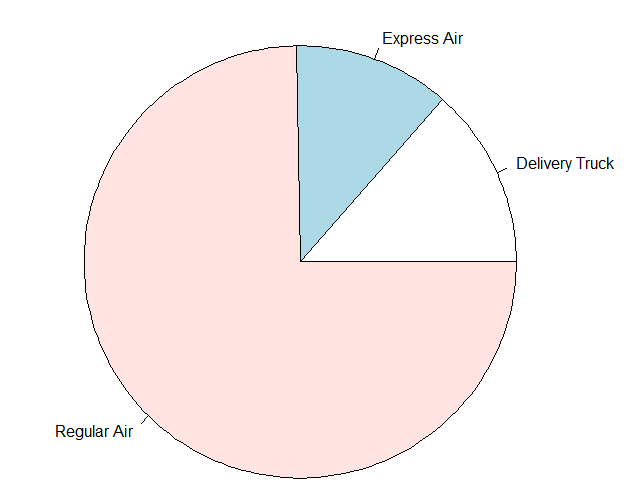
We are analyzing all the 4 independent variables from data set ‘Inventory’. The Ship Mode variable is the dependent variable. We perform Univariate and Bivariate analysis.

* **Order Quantity** follows a fixed saturated kind of distribution.
* **Sales** has right skewed distribution.
* The boxplot of **Sales** shows there are **outliers**.
* Almost 3/4th of the shipping modes belongs to Regular Air.
* Small Box takes more than 50% share among Product Containers.
* Paper has the largest share among Product Sub-Categories.







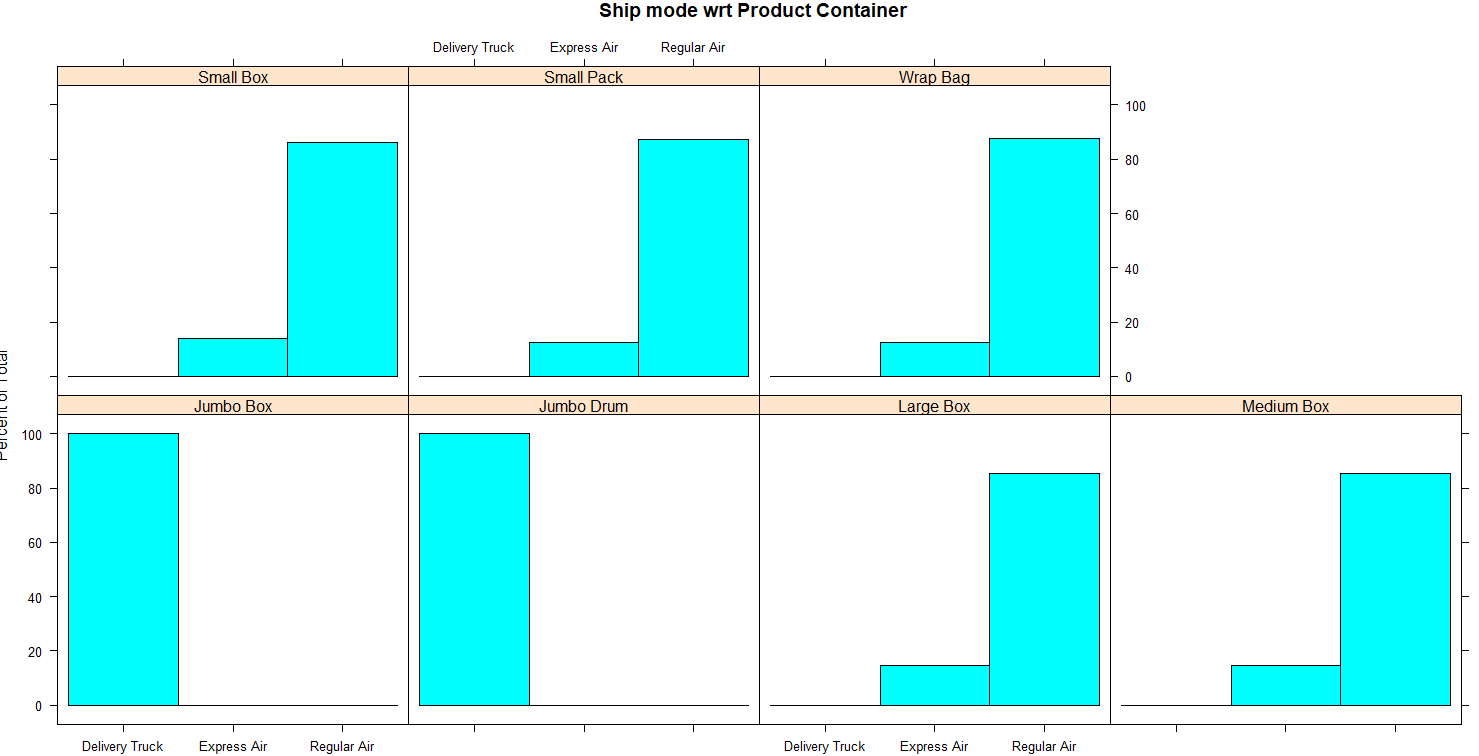


## Bi-Variate Analysis

We will analyze Ship Mode with the other variables from data set.

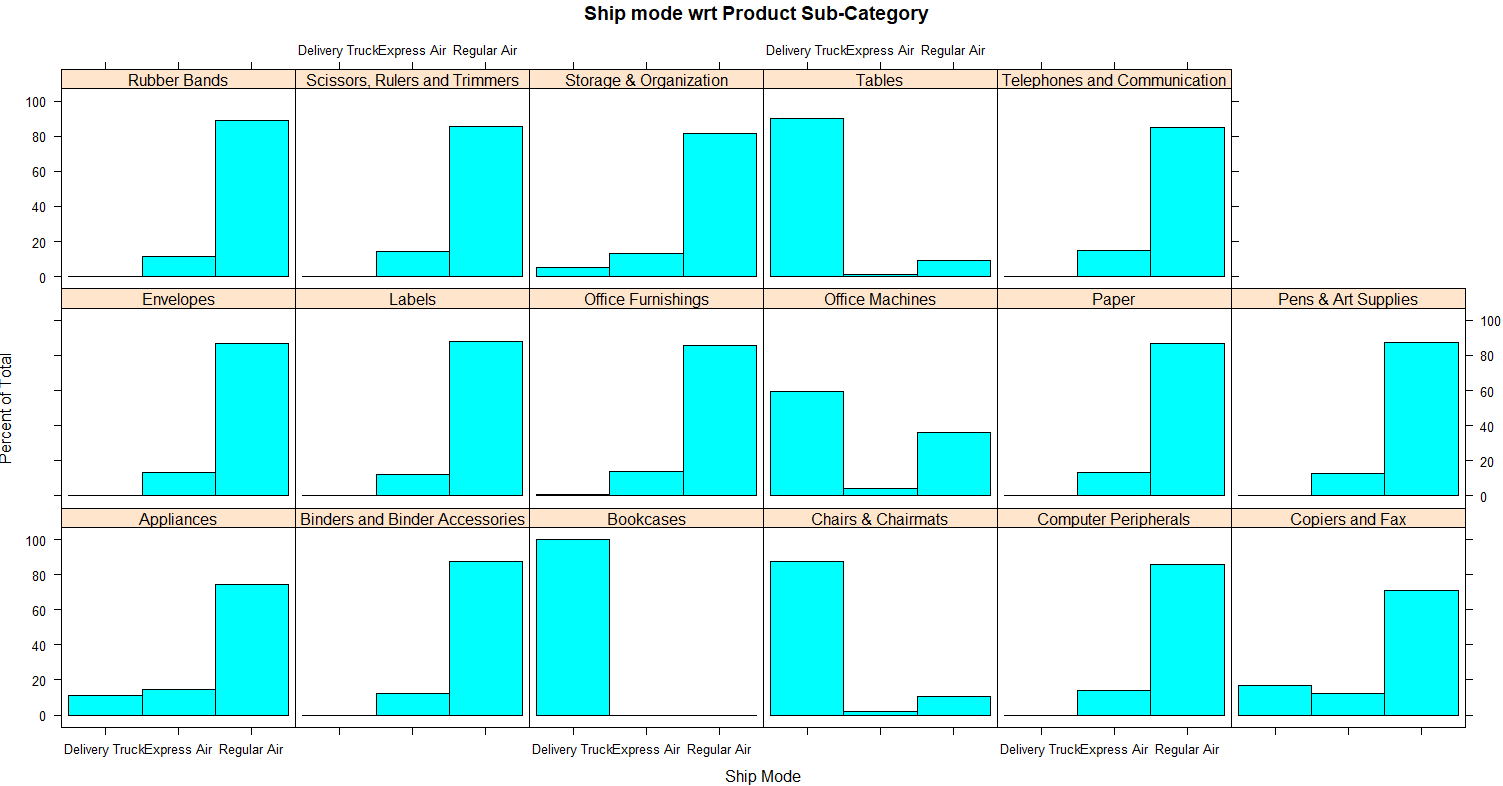
From the below plots we can say that out of 7 Product Container types Regular Air is the preferred shipping mode for 5 types except for Jumbo Box and Jumbo Drum for which shipping mode is Delivery Truck.

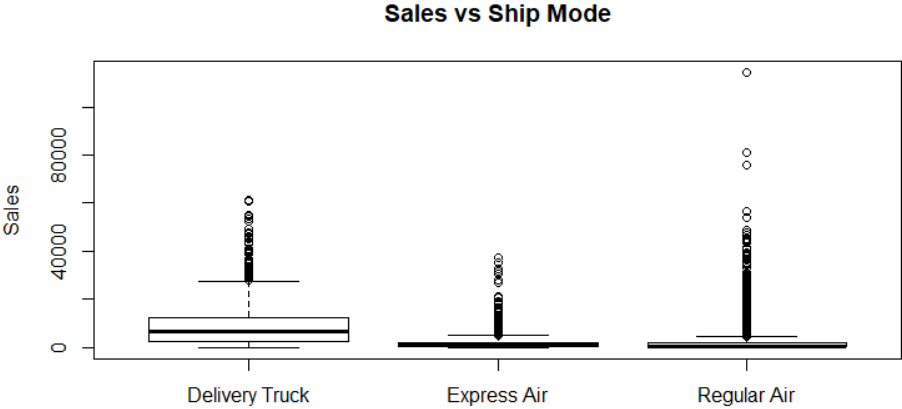
We infer that for delivering heaviest of packages, Delivery truck will be used. For all others use Regular Air followed by Express Air.



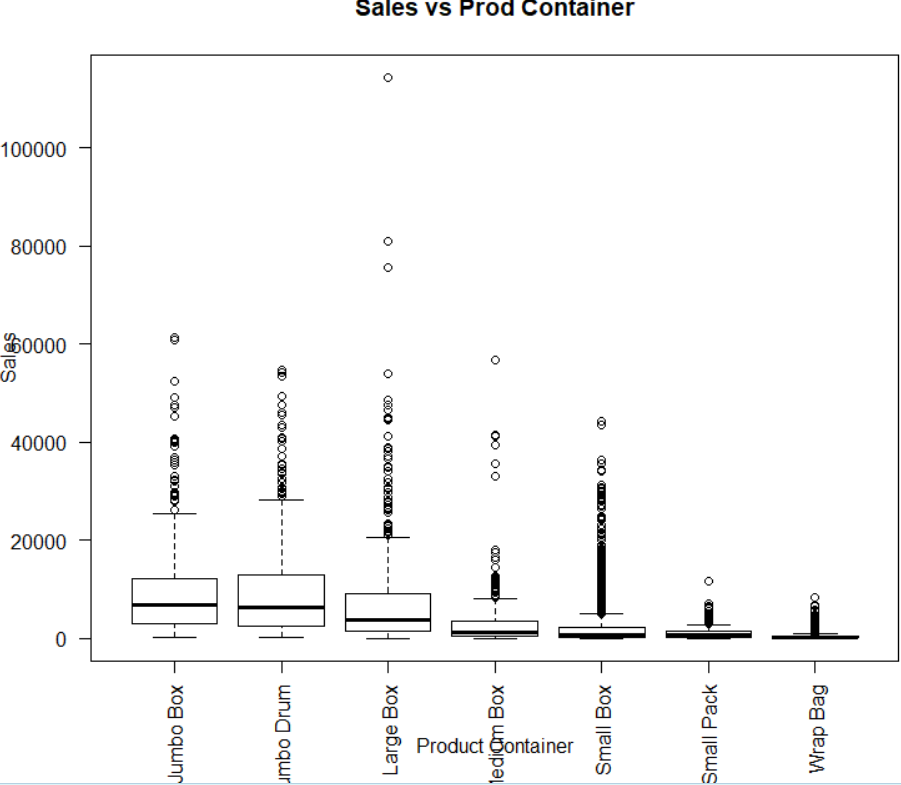
From below plots we can say that out of 17 Product Sub-Category types, 13 types are preferably delivered via Regular Air. For other types like Tables, Chairs and Chair mats, Bookcases and Office Machines Delivery truck is used.

One other insight that is easy to tell at this stage is there is no clear way to identify logic for Express Air as shipping mode. This might become problematic for the model.

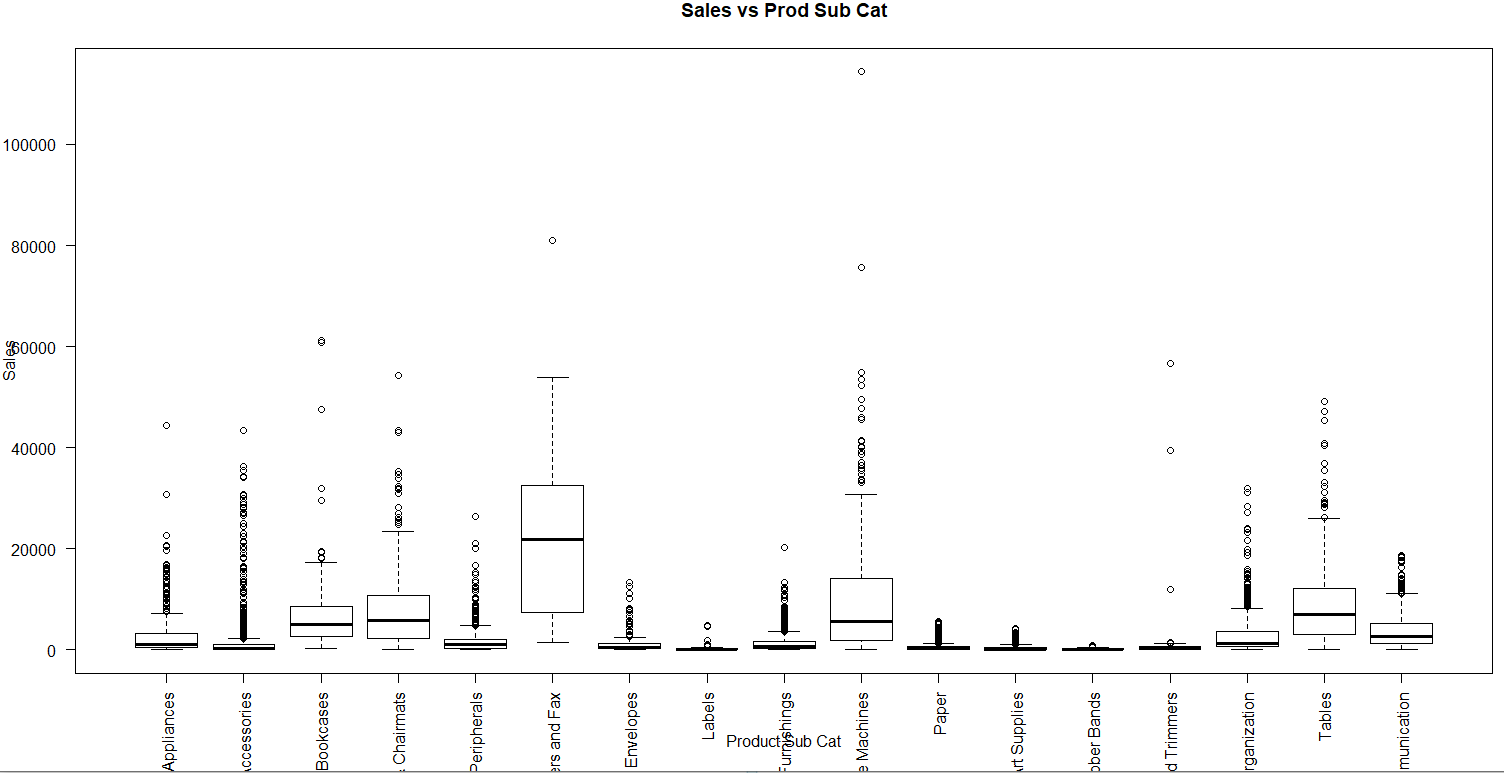




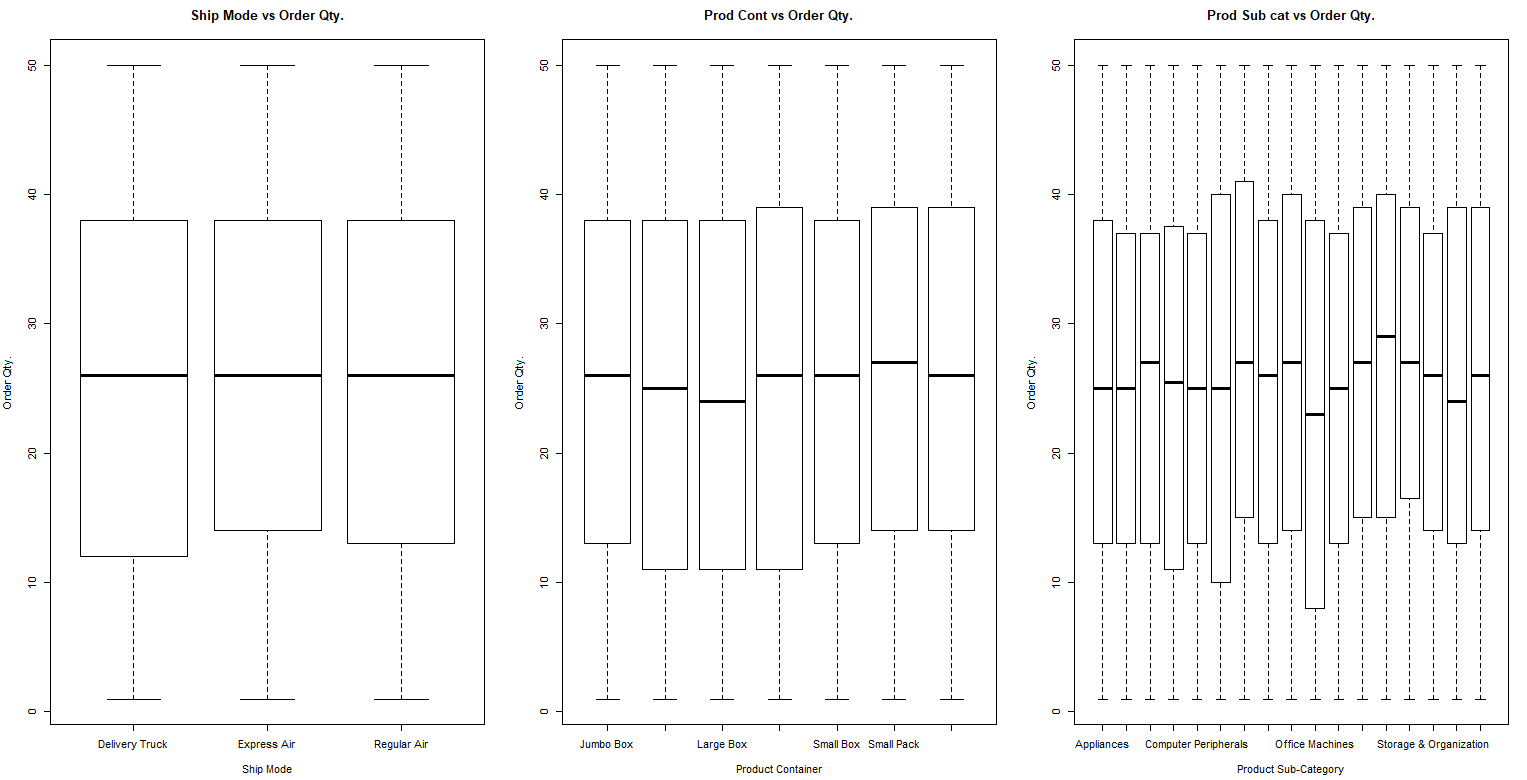
As per the plot above, median for Express Air and Regular Air Sales is nearly same. Comparatively Deliver Truck sales are higher that Express/Regular Air. But few sales of Regular Air are highest among all three shipping modes.



As per the plot above, Product Container types Jumbo Box, Jumbo Drum and Large amount to a big chunk of sales. Lowest contributors are Small Box, Small Pack and Wrap Bag. This seem usual as only small items(cheap) will be packed in Small containers. Few sales of Large Box items are highest among all container types.



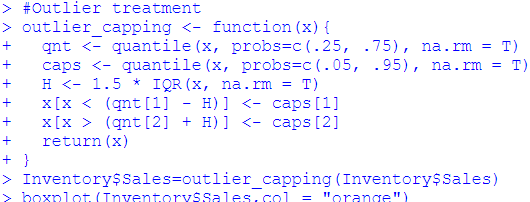
As per the plot above, Product Sub-Categories amounting to maximum sales are Copiers and Fax, Office Machines, Tables, Chairs-chair mats. Labels, Rubber Bands and Scissors, Rulers and Trimmers contribute least. Office machines and Copiers and Fax also have few individual sales which are highest among all.

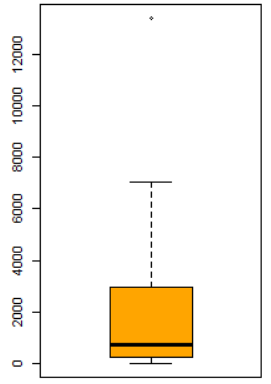


As per the plot above, Order Quantity is nearly same for all 3 shipping modes. Similarly Order Quantity for Product Container types and Product Sub-Category types is also consistent. Except this there is not much which can be said for above.

## Outlier Identification/Treatment

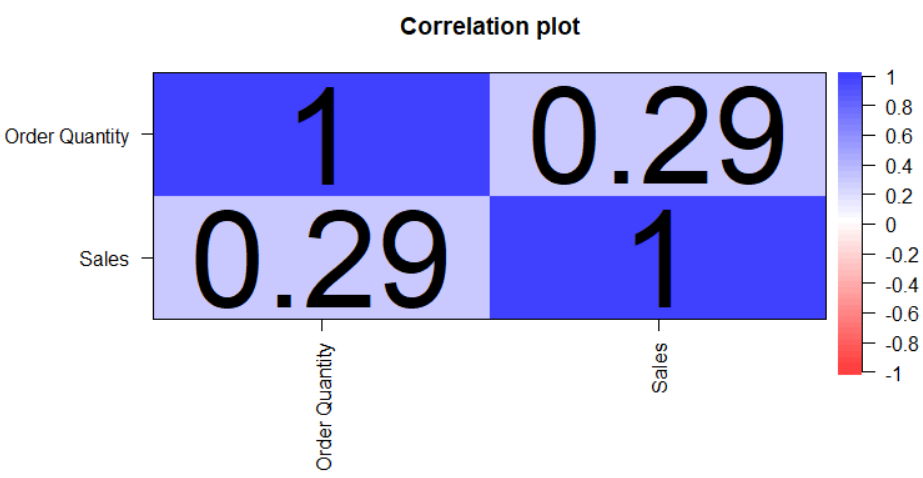
There are outliers in **Sales** variable which we saw in Univariate analysis done before. We do the outlier treatment using 5th and 95th percentile values using a custom-made function.

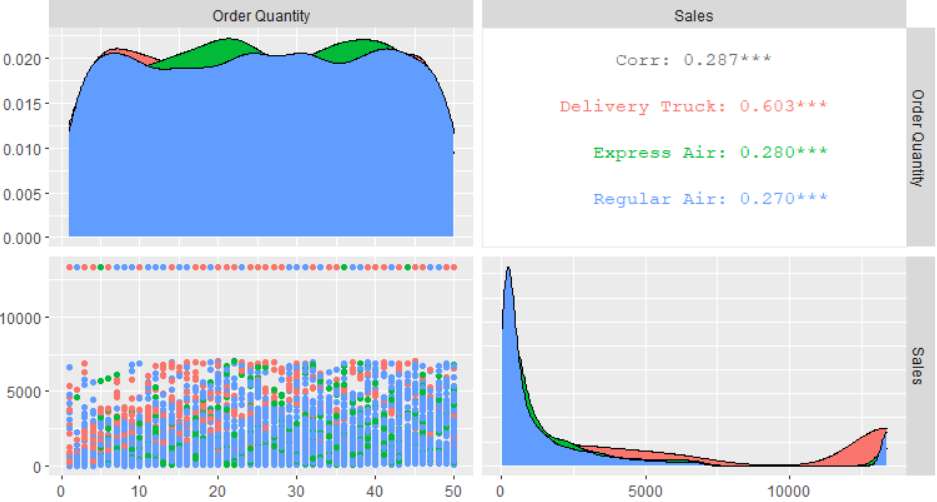




Please refer Appendix A for Source Code.

## Correlation/Multicollinearity





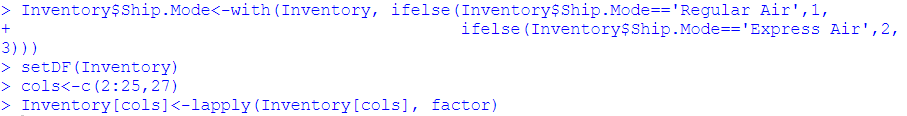
There seem no correlation among independent variables and as such there is no need to delve into this any further.

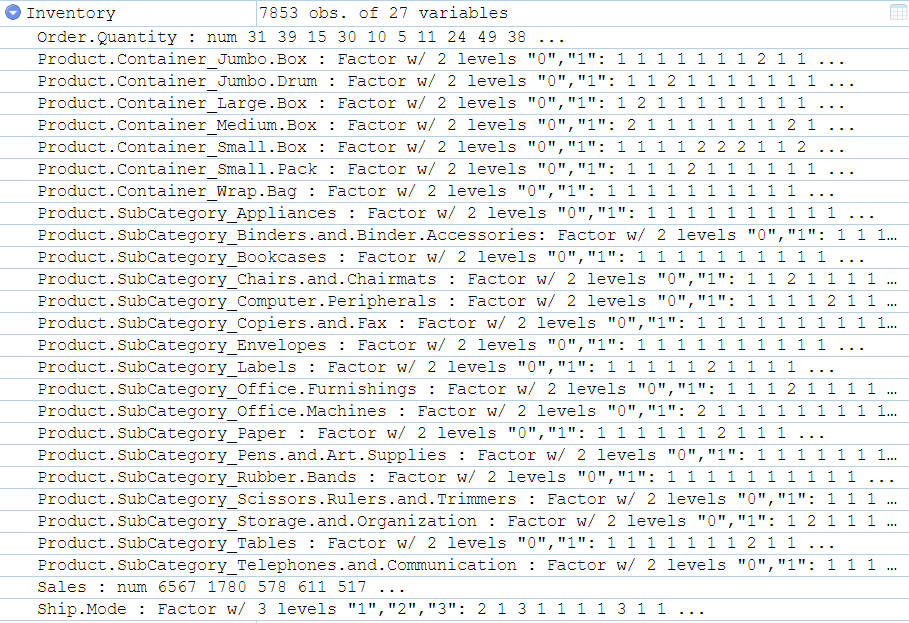
# ONE-HOT ENCODING

The dataset contains two independent variables and 1 target variable which are factors. So, we one-hot encode the independent variables. For target variable we do encoding so that 1 stand for Regular Air, 2 for Express Air and 3 for Delivery Truck.

We do above encoding because it allows representation of categorical data more expressive. Many machine learning algorithms cannot work with categorical data directly. The categories must be converted into numbers. This is required for both input and output variables that are categorical.





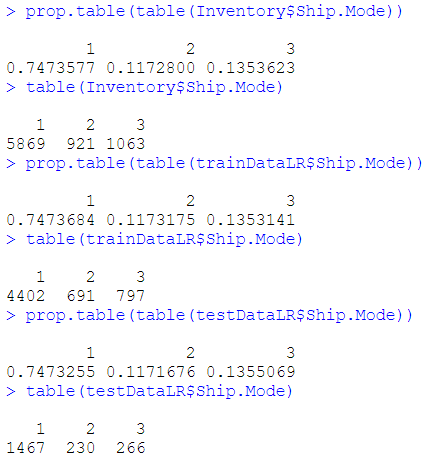


# Treating imbalance with ROSE

We oversample the minority classes i.e. Express Air and Delivery truck samples to treat the imbalance in our training data so that the model we build is accurate and robust. For this we use ROSE package in R (Random Over Sampling Examples). It helps us to generate artificial data based on sampling methods and smoothed bootstrap approach. This package has well defined accuracy functions to do the tasks quickly

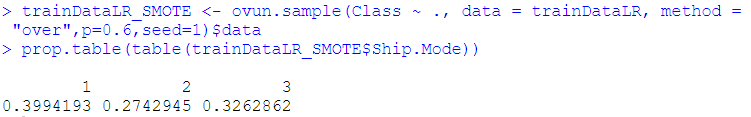
## Data Imbalance

We see that the data is highly imbalanced



## Oversample with ROSE

We increase the ratio of classes 1,2 and 3 from 75:12:14 to 40:27:33. We will build all models on ROSE data. In addition, for ensemble classifiers such as Decision Tree, Random Forest, Bagging, Boosting, Gradient Boosting models will be trained on imbalanced data as well because ensemble methods are known to handle imbalance pretty well.



# Logistic regression

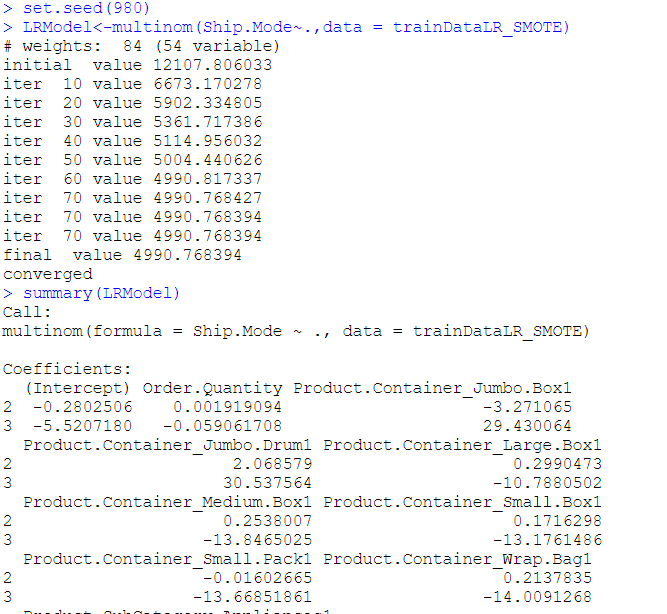
Logistic regression is part of the supervised learning. Here we use multinomial logistic regression. Multinomial regression is an extension of binomial logistic regression. The algorithm allows us to predict a categorical dependent variable which has more than two levels. Like any other regression model, the multinomial output can be predicted using one or more independent variable. The independent variables can be of a nominal, ordinal or continuous type..

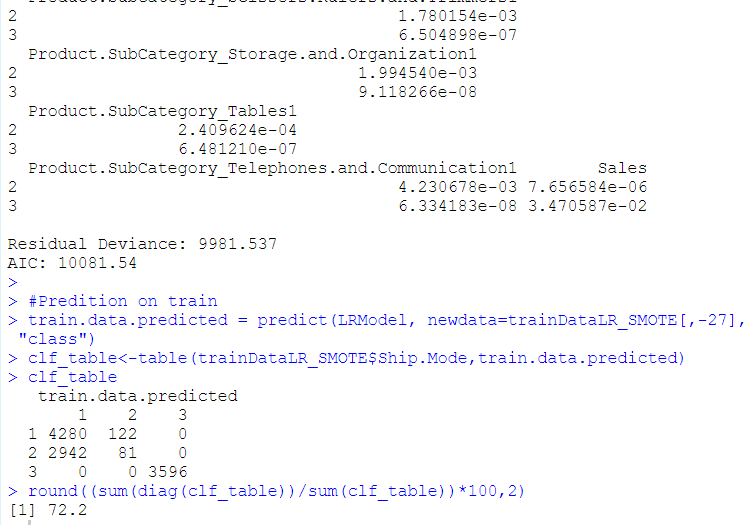
The independent variable, Ship Mode is multilevel. 1 is for Regular Air, 2 for Express Air and 3 for Deliver Truck shipping mode.

We can scale the data to reduce the impact of outliers. While model building, we have checked with scaled data as well but there was no impact on the model due to scaling. Hence, we are not scaling the data.

## Model Building

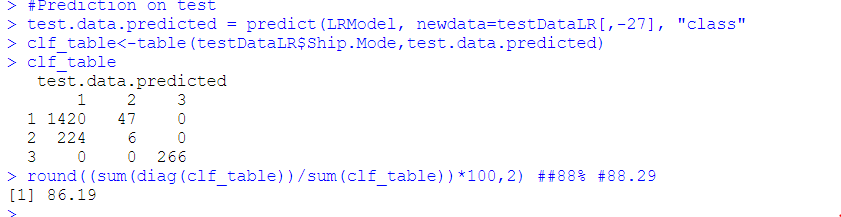
We will build with all the variables on the ROSE data and test the model on train data.

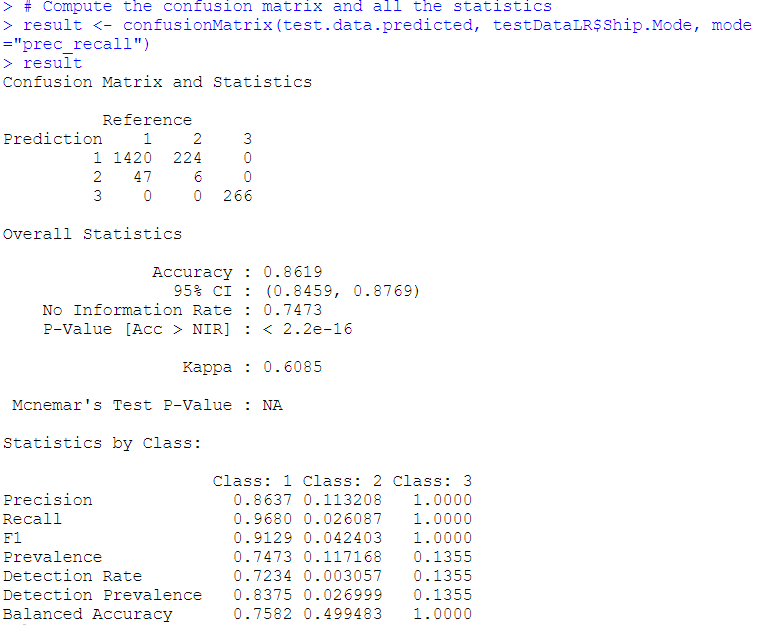




Please refer Appendix A for Source Code.

## Performance Metrics





|  |  |
| --- | --- |
| **Metrics** | **Value for Testing Dataset** |
| Precision | 0.86,0.11,1 |
| Recall | 0.97,0.02,1 |
| F1 | 0.91,0.04,1 |
| Balanced Accuracy | 0.75,0.5,1 |
| No. of Class 2 predicted correctly | 6/230 |

Please refer Appendix A for Source Code.

## Interpretation

Based on the performance metrics of the model we can make following inferences:

1. Overall model seems OK. No overfitting is observed.
2. Since there is class imbalance, we are considering F1 score as main accuracy metric. Class 1 and Class 3 have good F1 score. Class 2 has very low F1 score.
3. Model can predict only 6 Class 2 samples which is again on the lower side.
4. Testing accuracy is 86% which is decent.

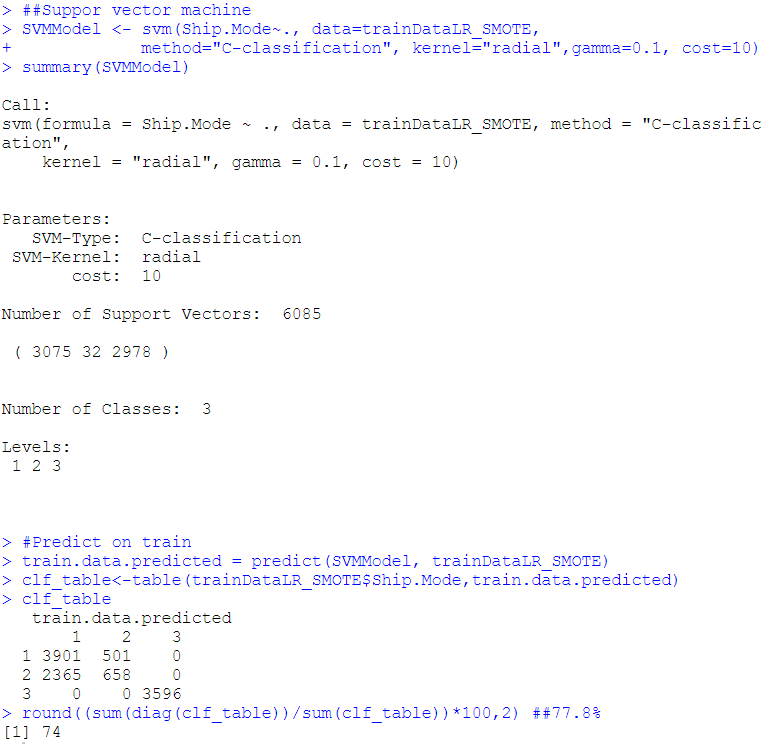
# Support Vector Machine

SVM performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels.

We also specify the type of usage we’d like for svm() with type=”C-classification” for classification (the default). The kernel argument has a variety of possible types including linear, polynomial, radial, and sigmoid. We use kernel=”radial” (the default) for this multi-class classification problem.

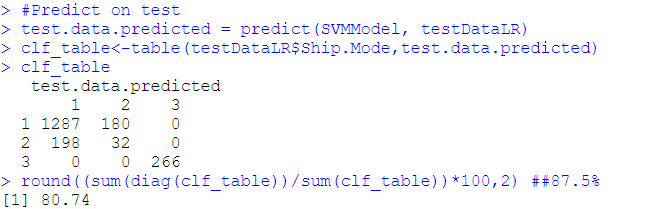
## Model Building

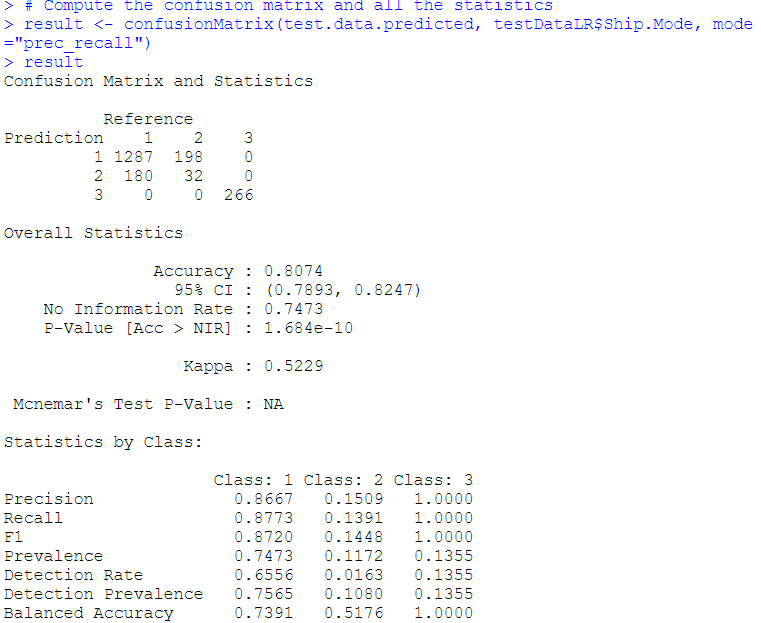
We build with all the variables on the ROSE data and test the model on train data.



Please refer Appendix A for Source Code.

## 7.2 Performance Metrics





|  |  |
| --- | --- |
| **Metrics** | **Value for Testing Dataset** |
| Precision | 0.87,0.15,1 |
| Recall | 0.88,0.14,1 |
| F1 | 0.87,0.14,1 |
| Balanced Accuracy | 0.74,0.52,1 |
| No. of Class 2 predicted correctly | 32/230 |

## Interpretation

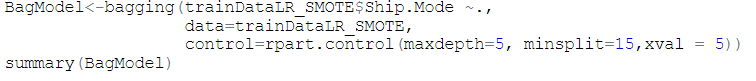
Based on the performance metrics of the model we can make following inferences:

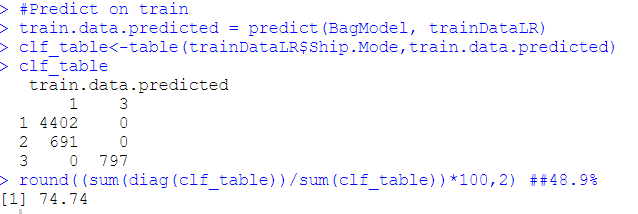
1. Overall model seems OK. No overfitting is observed.
2. This model is better than Multinomial Logistic Regression.
3. Class 1 and Class 3 have good F1 score. Class 2 has very low F1 score.
4. Model can predict 32 Class 2 samples which is bit low.
5. Testing accuracy is 81% which is decent.
6. **Bagging**

Bagging is also called as Bootstrap Aggregating. It is an ensemble machine learning algorithm designed to improve accuracy and stability of algorithm used in statistical classification and regression by reducing variance and avoiding overfitting.

* 1. **Model Building**

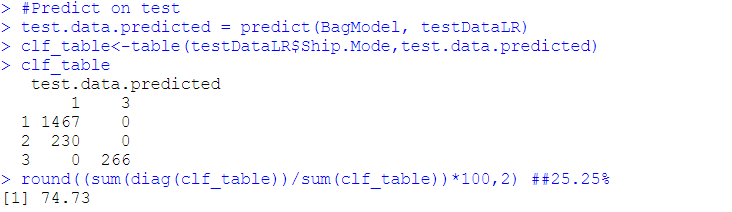
We build with all the variables on the imbalanced data and test the model on train data.

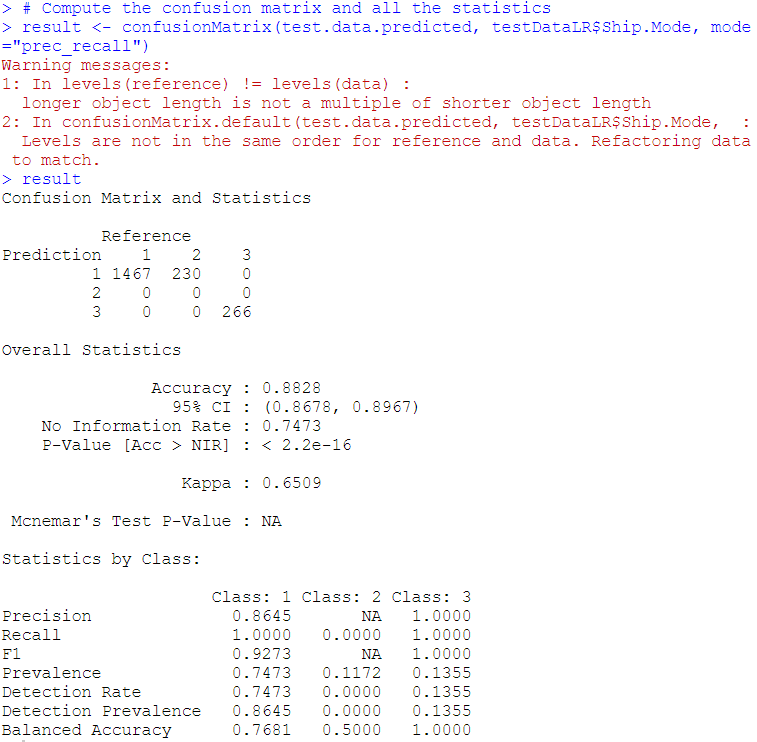




Please refer Appendix A for Source Code.

* 1. **Performance Metrics**





|  |  |
| --- | --- |
| **Metrics** | **Value for Testing Dataset** |
| Precision | 0.86,NA,1 |
| Recall | 1,0,1 |
| F1 | 0.92,NA,1 |
| Balanced Accuracy | 0.77,0.5,1 |
| No. of Class 2 predicted correctly | 0 |

* 1. **Interpretation**

Based on the performance metrics of the model we can make following inferences:

* + 1. No overfitting is observed.
    2. This model predicts Class 1 and 3 accurately but can’t predict Class 2 at all.
    3. Class 1 and Class 3 have good F1 score.
    4. This model has to be rejected.
    5. Testing accuracy is 88% which is decent but is of no use.

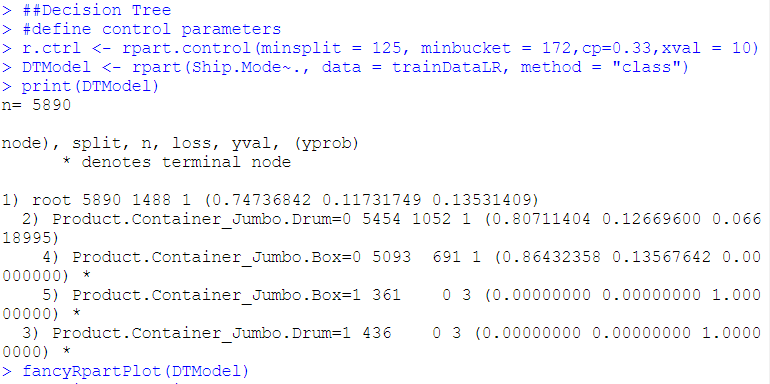
1. **Decision Tree**

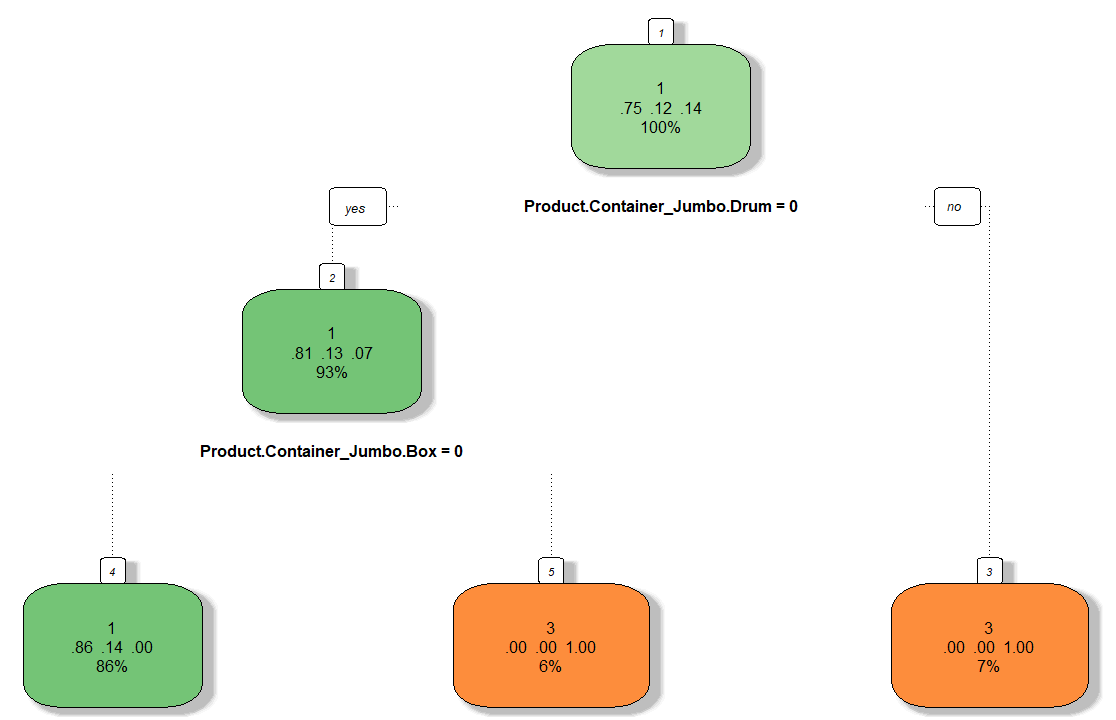
Decision Tree is a Supervised Machine Learning algorithm which looks like an inverted tree, wherein each node represents a predictor variable (feature), the link between the nodes represents a Decision and each leaf node represents an outcome (response variable).

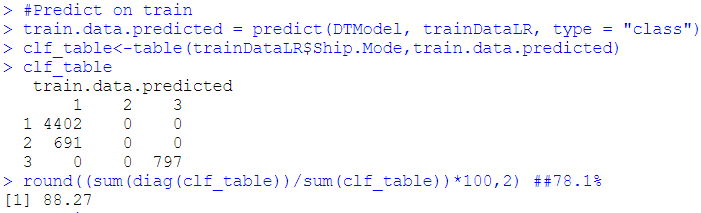
The algorithm of the decision tree models works by repeatedly partitioning the data into multiple sub-spaces so that the outcomes in each final sub-space is as homogeneous as possible. This approach is technically called recursive partitioning. The produced result consists of a set of rules used for predicting the outcome variable.

* 1. **Model Building**

We build with all the variables on the imbalanced data and test the model on train data.

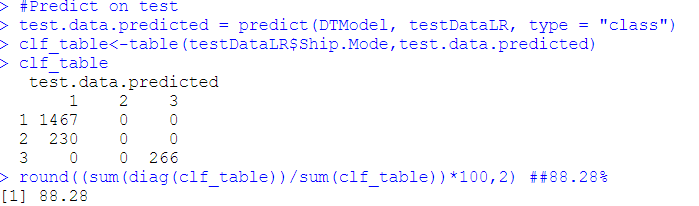


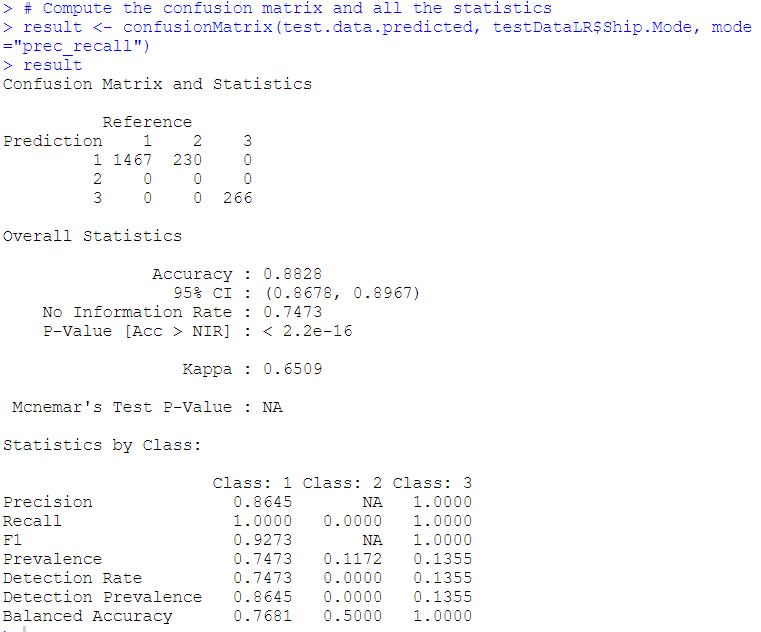




Please refer Appendix A for Source Code.

* 1. **Performance Metrics**





|  |  |
| --- | --- |
| **Metrics** | **Value for Testing Dataset** |
| Precision | 0.86,NA,1 |
| Recall | 1,0,1 |
| F1 | 0.93,NA,1 |
| Balanced Accuracy | 0.77,0.5,1 |
| No. of Class 2 predicted correctly | 0 |

* 1. **Interpretation**

Based on the performance metrics of the model we can make following inferences:

* + 1. No overfitting is observed.
    2. This model predicts Class 1 and 3 accurately but can’t predict Class 2 at all.
    3. Class 1 and Class 3 have good F1 score.
    4. Like Bagging, this model has to be rejected as well.
    5. Testing accuracy is 88% which is decent but is of no use.

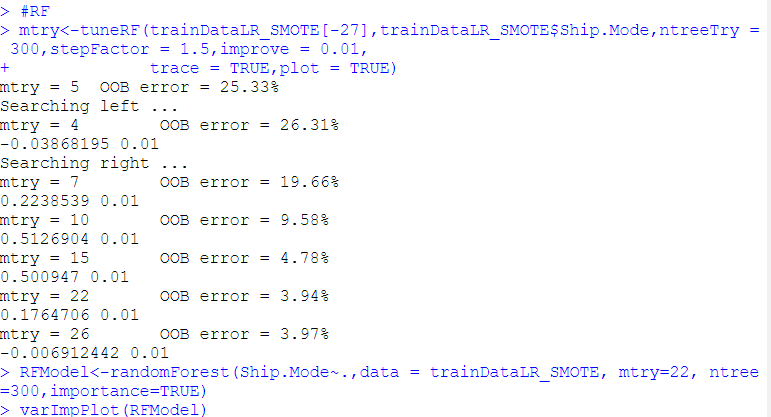
# Random Forest

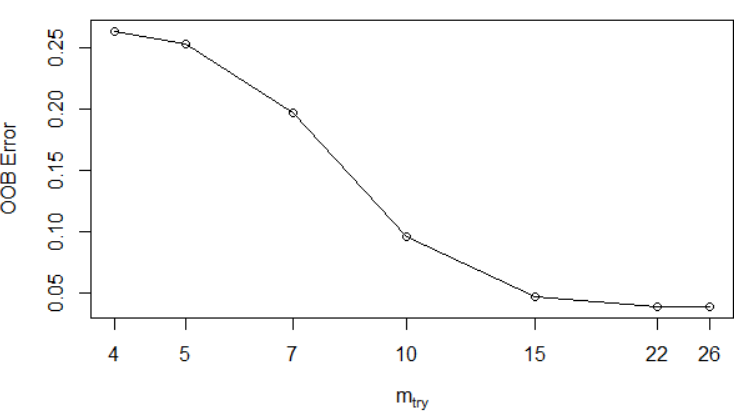
Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction.

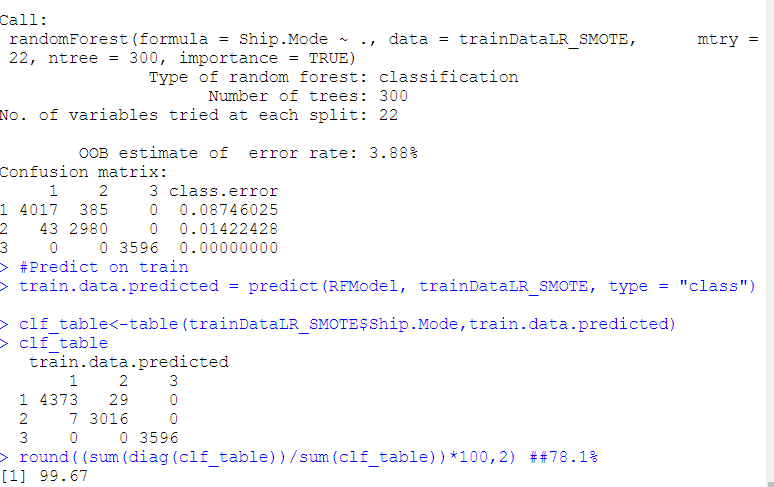
The random forest is a classification algorithm consisting of many decision trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

## 10.1 Model Building

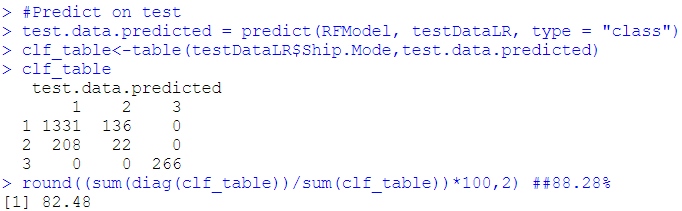
We build with all the variables on the ROSE data and test the model on train data.

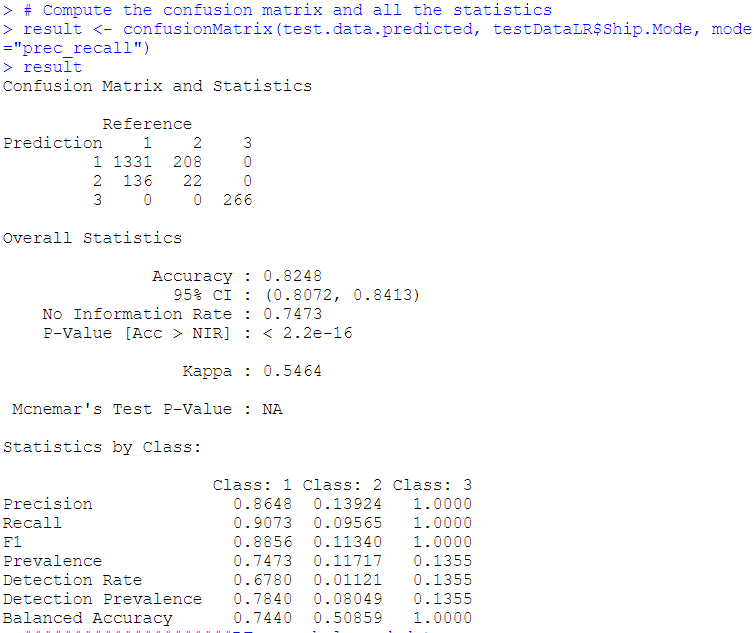






## 10.2 Performance Metrics





|  |  |
| --- | --- |
| **Metrics** | **Value for Testing Dataset** |
| Precision | 0.86,0.13,1 |
| Recall | 0.91,0.09,1 |
| F1 | 0.89,0.11,1 |
| Balanced Accuracy | 0.74,0.51,1 |
| No. of Class 2 predicted correctly | 22/230 |

## 10.3 Interpretation

Based on the performance metrics of the model we can make following inferences:

* + 1. Looks like model has overfitted. Has almost 100% training accuracy.
    2. This model is comparable to SVM. It’s way better than all other models.
    3. Class 1 and Class 3 have good F1 score. Class 2 has very low F1 score.
    4. Model can predict 22 Class 2 samples which is bit low.
    5. Testing accuracy is 82% which is decent but is o no use due to overfitting.

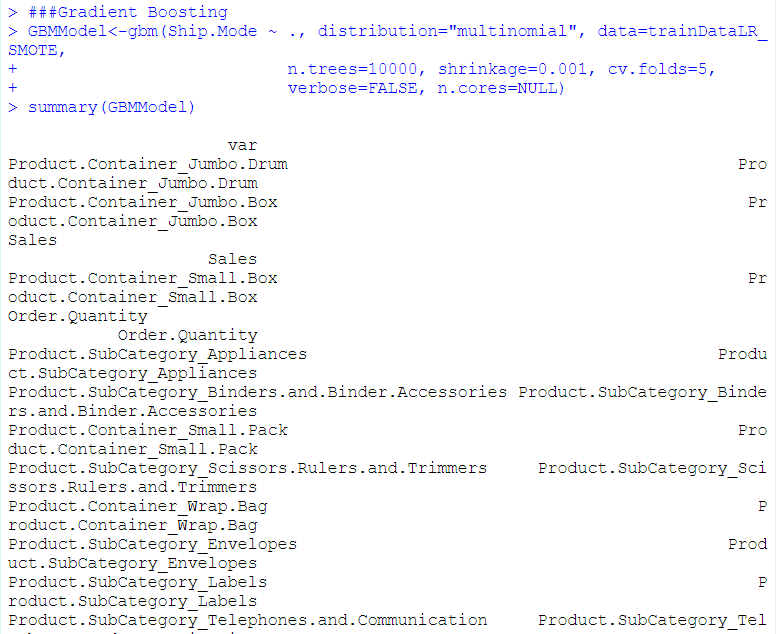
# Gradient Boosting

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model.

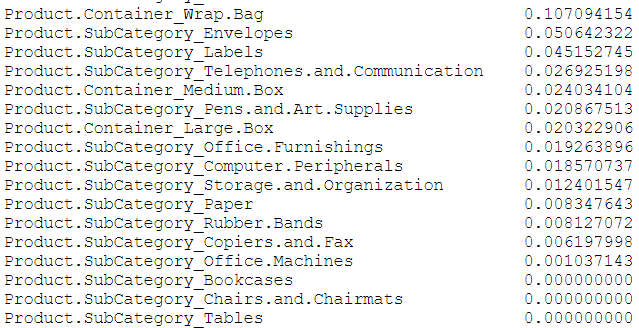
Whereas random forests build an ensemble of deep independent trees, GBMs build an ensemble of shallow and weak successive trees with each tree learning and improving on the previous. When combined, these many weak successive trees produce a powerful “committee” that are often hard to beat with other algorithms.

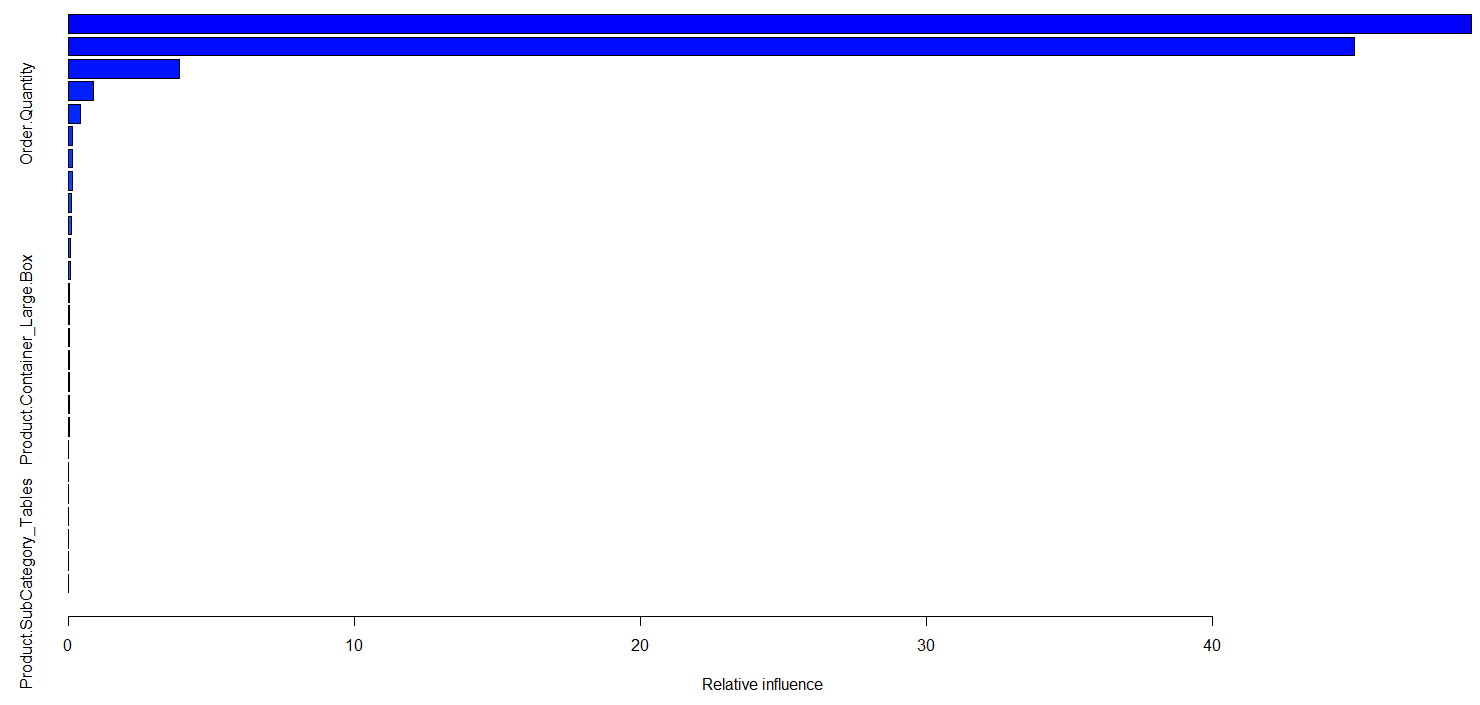
## Model Building

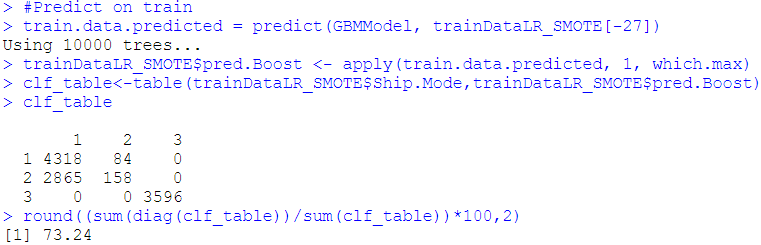
We build with all the variables on the ROSE data and test the model on train data.



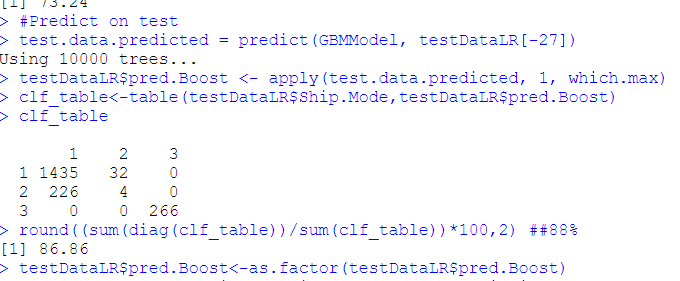


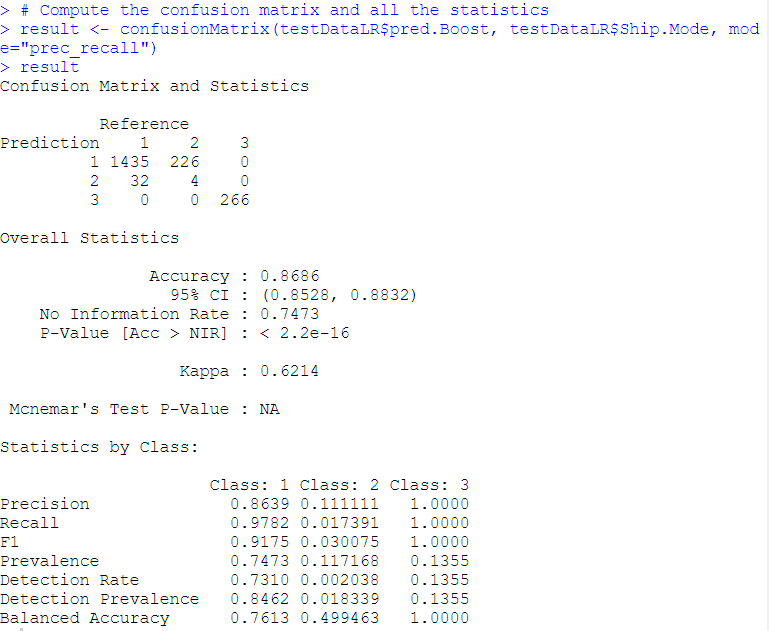






## Performance Metrics





|  |  |
| --- | --- |
| **Metrics** | **Value for Testing Dataset** |
| Precision | 0.86,0.11,1 |
| Recall | 0.98,0.17,1 |
| F1 | 0.92,0.03,1 |
| Balanced Accuracy | 0.76,0.5,1 |
| No. of Class 2 predicted correctly | 4/230 |

## Interpretation

Based on the performance metrics of the model we can make following inferences:

* + 1. Overall model seems OK. No overfitting is observed.
    2. This model is comparable to SVM and Random Forest. Better than all other models.
    3. Class 1 and Class 3 have good F1 score. Class 2 has very low F1 score.
    4. Model can predict 4 Class 2 samples which is too low.
    5. Testing accuracy is 87% which is decent.

1. **Model Comparison**

* Bagging and Decision Tree models are outright rejected due to their inability to predict Class 2 samples.
* All the models including base and ensemble classifiers were able to get pretty good F1 scores for Class 1 and Class 3 samples.
* All the models struggled to get a moderate F1 score for Class 2 samples. This once again highlights our insight derived from EDA that shipping mode Express Air is difficult to predict because there seems hardly any logic among independent variables. Even complex ML algorithms are finding it hard to find out some pattern w.r.t Express Air.
* Only two models SVM and RF were able to predict some number of Class 2 samples.
* Out of the above two we decide to select **SVM as our final predictive model**.
* We may look to increase the number of predictions of Class 2 samples if possible, with Hyper parameter tuning.
* We also did undersampling and combination of over/undersampling but in those approaches also it was difficult to increase the predictive power of models for classifying Class 2 samples.

1. **Conclusion**

We have built various models to understand the factors which influence the choice of Shipping mode. Using models like RF and Gradient Boosting we found out that most important factors are:

Sales

Order Quantity

Product Container Type – Jumbo Box

Product Container Type – Jumbo Drum

The model built using SVM multi class classifier is the best model as testing accuracy is about 81% and it is able to predict 32 out of 230 test samples of Shipping mode Express Air. The model seems quite stable.

Key insights are –

1. For predicting shipping modes – Regular Air and Delivery Truck above given 4 factors are good enough.
2. Predicting shipping mode – Express Air is herculean task as evident from both EDA and model building. There is just not enough predictive pattern.

Advice needs to be given to the management to chalk out a strategy for selecting shipping mode as Express Air. There is just no logic, pattern in how items are being delivered via this mode which should not be the case in a Supply Chain and Logistics firm. This seem to be the biggest point which comes out of this analysis. We need to tell management that you are not doing or following any procedure for selecting shipping mode as Express Air. Without inducing any strategy/logic related to Express Air it is impossible for ML models to correctly predict shipping mode as Express Air.

1. **Appendix A – Source Code**

****