Instacart Market Basket Analysis

An edWisor.com’s project

Submitted by: Sandhya Parkar

Table of Contents

Contents

Instacart Market Basket Analysis

1.Problem Statement:

To predict which products customer will buy again.

2.Business understanding:

Instacart, a grocery ordering and delivery app, aims to make it easy to fill your refrigerator and pantry with your personal favorites and staples when you need them. After selecting products through the Instacart app, personal shoppers review your order and do the in-store shopping and delivery for you.

In this competition, Instacart is challenging the Kaggle community to use this anonymized data on customer orders over time to predict which previously purchased products will be in a user's next order.

3.Data Understanding:

We have six sets of data of which are as described below:

1. aisles : This data set contains aisle_id and aisle variables.
2. departments : It contains department_id and department variables.
3. order_products_prior : It contains 4 variables and 32434489 observations.
4. order_products_train : It contains 1384617 observations and 4 variables.
5. orders : It contains 3421083 observations and 7 variables.
6. products : It contains 49688 observations and 4 variables.

4. Data Preprocessing:

Our data contains six tables on which we performed some preprocessing like some of the variables are reshaped using as.factor like variables aisle, department, eval_set of orders table and products_name variable of products table.Here eval_set of orders table contains train set , test set and prior set of data. We also performed some visualization to look at the proportion of data to these particular sets. Here Products table contains aisle_id and department_id so we joined products table with aisle and department tables on the basis of unique keys. Similarly order_products_prior and order_products_train contain same variables so we merged two tables into one which is later joined with orders table. In this way we got a single table order_products_prior to deal with.

5. Feature Engineering:

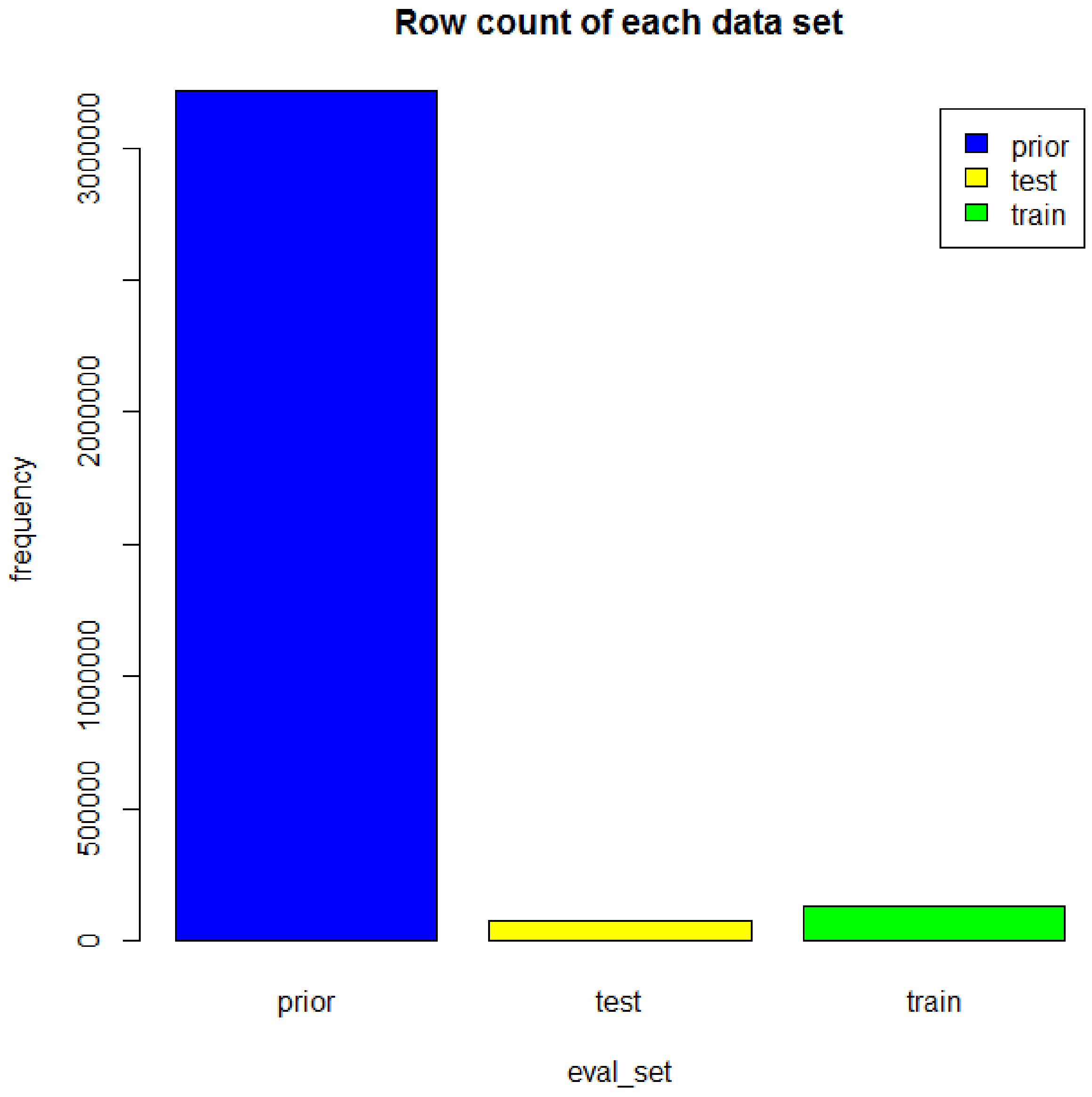
The predictive power of a model can be increased significantly if supplied with the right feature engineered variable inputs. Here we have huge data sets and a lot of scope to create new features and also some new tables which are later merged in a single table on the basis of unique key.

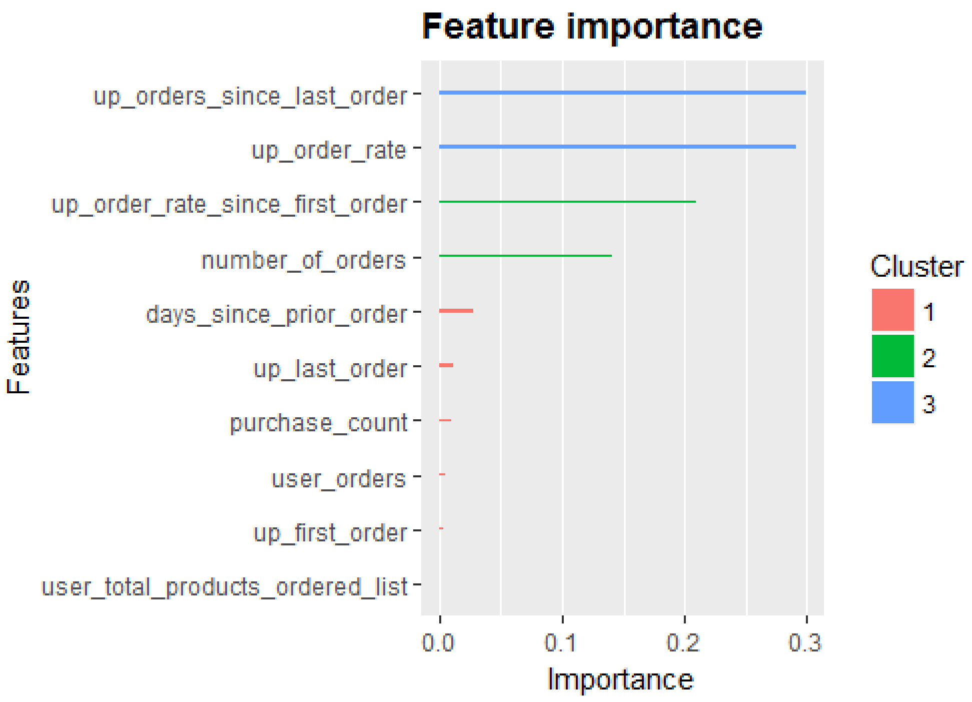
We created new features like number_of_orders, user_order, first_order, next_order, last_order, purchase_count of a particular user_id, user_total_products_ordered, uniq_prod etc. and similarly some new tables like products_summ, user_details, user_prod_details and many more. Lastly we combined all of these tables into a single table which then splitted into train data set and test data set.

After splitting into testing and training frame we removed some unimportant variables.

6. Visualization:

We performed visualization using bar plot to view the row count of eval_set type in orders data set and also for looking at the important feature we plot a graph between various features used in our modelling process.




Feature importance
Features
up_orders_since_last_order
up_order_rate
up_order_rate_since_first_order
number_of_orders
days_since_prior_order
up_last_order
purchase_count
user_orders
up_first_order
user_total_products_ordered_list
0.0
0.1
0.2
0.3
Importance
Cluster
1
2
3

7. Predictive Modelling:

We used xgboost (Extreme Gradient Boosting) for predictive modelling. Xgboost algorithm is used for faster processing and memory optimization as here we are dealing with very huge data set.

Our model was able to predict target variable with AUC 99.8.

Accuracy of our model is 98%.

We also find the importance of features using xgboost.ggplot.importance() method.

7. Conclusion:

To predict whether a consumer will purachase a product again, we performed various techniques on our data. We have performed preprocessing and feature engineering on our data. We created near about 33 features. We tested various models for our data to provide better accuracy and our finalized algorithm for model building is xgboost (Xtreame Gradient Boosting Algorithm). Applying this model we obtained a mean accuracy of 98%.

Thank You