InstacartFinalProject

Shivangi Vashi

6/23/2020

ALY 6040 Data Mining Applications

Instacart Final Project

Shivangi Vashi

Yihong Qiu

Md Tajrianul Islam

Instructor: Kasun Samarasinghe

Spring 2020

June 23 2020

Northeastern University

Introduction

Instacart is a grocery delivery application that works in the following way- users can select products through the app, and then personal shoppers review the order, shop in-store and deliver the products to the users. We got the data from Instacart's Market Basket Analysis on kaggle.com here (https://www.kaggle.com/c/instacart-market-basket-analysis/overview).

The Instacart data set is anonymized and contains samples of over 3 million grocery orders from 200,000+ Instacart users.

The reason we chose this data is because it is varied and some very clear applications for resolving business questions. It will allow us to perform different data mining algorithms and analyze it in different ways. It also has time information with enough granularity, so the option of some time series analysis is also there.

In this final project, there are four parts of the Instacart Market Basket analysis. The first part is EDA. We explore the frequency of reordering, when in a day or week do users order the most, popular products, and so on. This gives us an idea about user buying patterns. The other parts is using different methods to solve the business problem.

Business Problem:

On its website Instacart has a recommendation system, that suggests the users some items to buy again. Our task is to predict which items will be reordered on the next order.

We approach this using 3 methods: -Logistic Regression: We use logistic regression to predict if a product will be reordered or not. Since the 'reordered' variable is binary, this is a classification problem that can be solved using logistic regression

-Association Rule Mining: We use the apriori algorithm perform market basket analysis, to generate association rules with different values of support and confidence, to predict what products are closely associated with or frequently bought with what products

-Gradient Boosting

Reading the dataset,importing relevant libraries.

Data Wrangling

Data Preparation

Since the dataset is very large, with Prior orders having 32 million rows, we subset the data to reduce calculation time. We did this by randomly sampling users, then only keeping their orders and prior order information by performing inner joins with the order prior and train datasets.

```
## Number of orders (before):  3421083
```

```
## Number of orders (after):  346739
```

```
## [1] 3268618       4
```

```
## [1] 137284      4
```

```
## [1] 3061150      10
```

```
## [1] 137284     10
```

```
##    order_id user_id eval_set order_number order_dow order_hour_of_day
## 10  3106101      18    prior            2         0                17
## 11  3106101      18    prior            2         0                17
## 12  3106101      18    prior            2         0                17
## 13  3106101      18    prior            2         0                17
## 14  3106101      18    prior            2         0                17
## 15  1860960      18    prior            3         1                19
##    days_since_prior_order product_id add_to_cart_order reordered
## 10                      1      36216                 1         1
## 11                      1       4461                 2         0
## 12                      1       5876                 3         0
## 13                      1        810                 4         0
## 14                      1      31717                 5         0
## 15                      8      36216                 1         1
##   order_id user_id eval_set order_number order_dow order_hour_of_day
## 1  2461523      18    train            7         6                 9
## 2  2461523      18    train            7         6                 9
## 3  2461523      18    train            7         6                 9
## 4  2461523      18    train            7         6                 9
## 5  2461523      18    train            7         6                 9
## 6  2461523      18    train            7         6                 9
##   days_since_prior_order product_id add_to_cart_order reordered
## 1                      7      36216                 1         1
## 2                      7      47546                 2         1
## 3                      7      21137                 3         1
## 4                      7       5450                 4         0
## 5                      7       8518                 5         0
## 6                      7      22031                 6         1
```

```
##    aisle_id                      aisle
## 1:        1      prepared soups salads
## 2:        2          specialty cheeses
## 3:        3        energy granola bars
## 4:        4              instant foods
## 5:        5 marinades meat preparation
## 6:        6                      other
```

```
##    department_id    department
## 1:             1        frozen
## 2:             2         other
## 3:             3        bakery
## 4:             4       produce
## 5:             5       alcohol
## 6:             6 international
```

```
##   order_id product_id add_to_cart_order reordered
## 1        3      33754                 1         1
## 2        3      24838                 2         1
## 3        3      17704                 3         1
## 4        3      21903                 4         1
## 5        3      17668                 5         1
## 6        3      46667                 6         1
```

```
##   order_id product_id add_to_cart_order reordered
## 1      719      45683                 1         0
## 2     1865      33129                 1         1
## 3     1865      31553                 2         0
## 4     1865      29487                 3         0
## 5     1865      47734                 4         0
## 6     1865       3188                 5         0
##    order_id user_id eval_set order_number order_dow order_hour_of_day
## 1:  2780889      18    prior            1         6                18
## 2:  3106101      18    prior            2         0                17
## 3:  1860960      18    prior            3         1                19
## 4:  3133044      18    prior            4         0                16
## 5:  1020460      18    prior            5         2                16
## 6:   441977      18    prior            6         6                12
##    days_since_prior_order
## 1:                     NA
## 2:                      1
## 3:                      8
## 4:                      6
## 5:                      9
## 6:                      4
```

```
##    product_id                                                      product_name
## 1:          1                                         Chocolate Sandwich Cookies
## 2:          2                                                   All-Seasons Salt
## 3:          3                               Robust Golden Unsweetened Oolong Tea
## 4:          4 Smart Ones Classic Favorites Mini Rigatoni With Vodka Cream Sauce
## 5:          5                                          Green Chile Anytime Sauce
## 6:          6                                                       Dry Nose Oil
##    aisle_id department_id
## 1:       61            19
## 2:      104            13
## 3:       94             7
## 4:       38             1
## 5:        5            13
## 6:       11            11
```

```
##    aisle_id       aisle
##   "integer" "character"
```

```
## department_id    department
##     "integer"   "character"
```

```
##          order_id        product_id add_to_cart_order         reordered
##         "integer"         "integer"         "integer"         "integer"
```

```
##          order_id        product_id add_to_cart_order         reordered
##         "integer"         "integer"         "integer"         "integer"
```

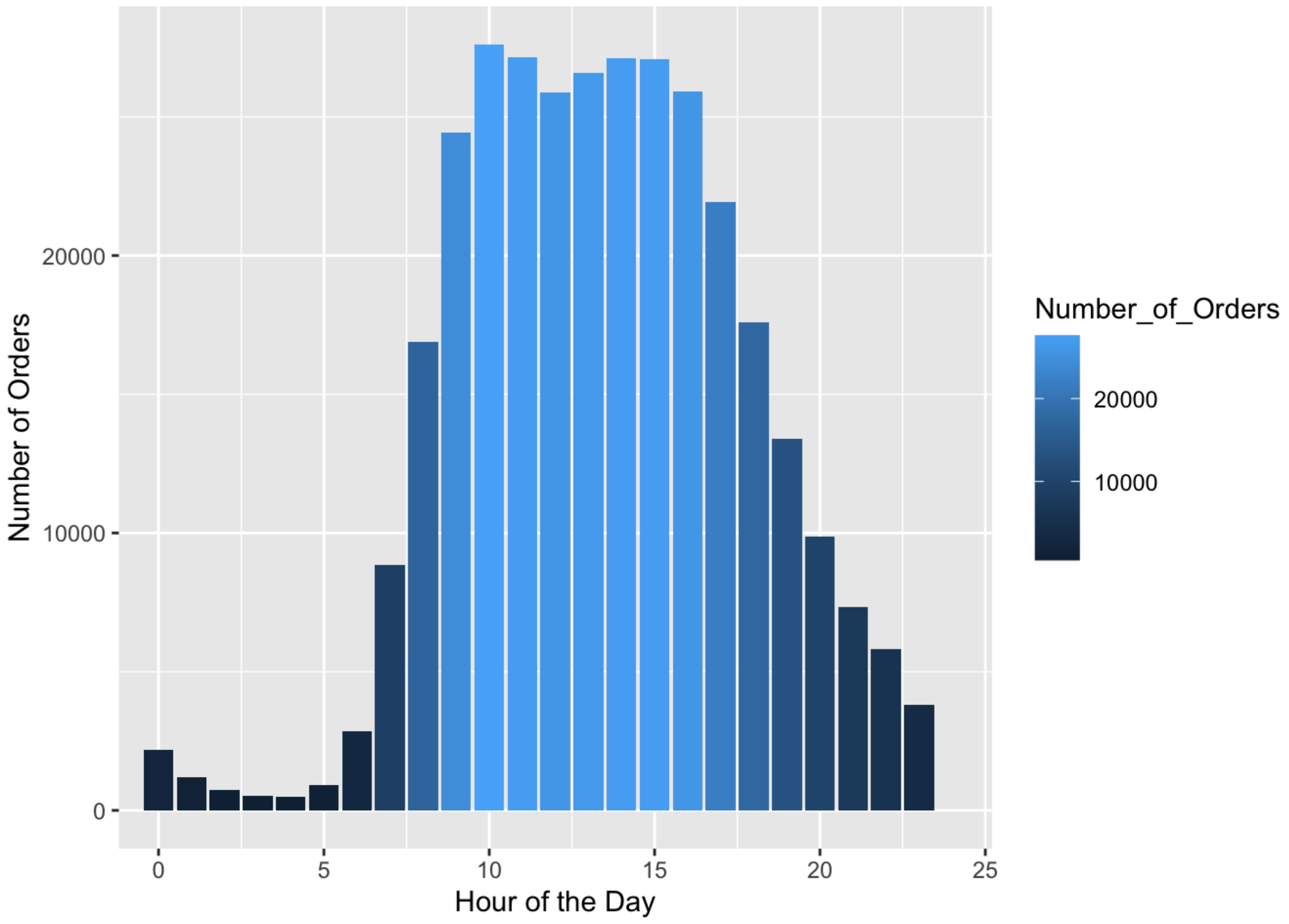
```
##               order_id                user_id               eval_set
##              "integer"              "integer"            "character"
##           order_number              order_dow      order_hour_of_day
##              "integer"              "integer"              "integer"
## days_since_prior_order
##              "numeric"
##   order_id user_id eval_set order_number order_dow order_hour_of_day
## 1  3106101      18    prior            2         0                17
## 2  1860960      18    prior            3         1                19
## 3  3133044      18    prior            4         0                16
## 4  1020460      18    prior            5         2                16
## 5   441977      18    prior            6         6                12
## 6  2461523      18    train            7         6                 9
##   days_since_prior_order
## 1                      1
## 2                      8
## 3                      6
## 4                      9
## 5                      4
## 6                      7
```

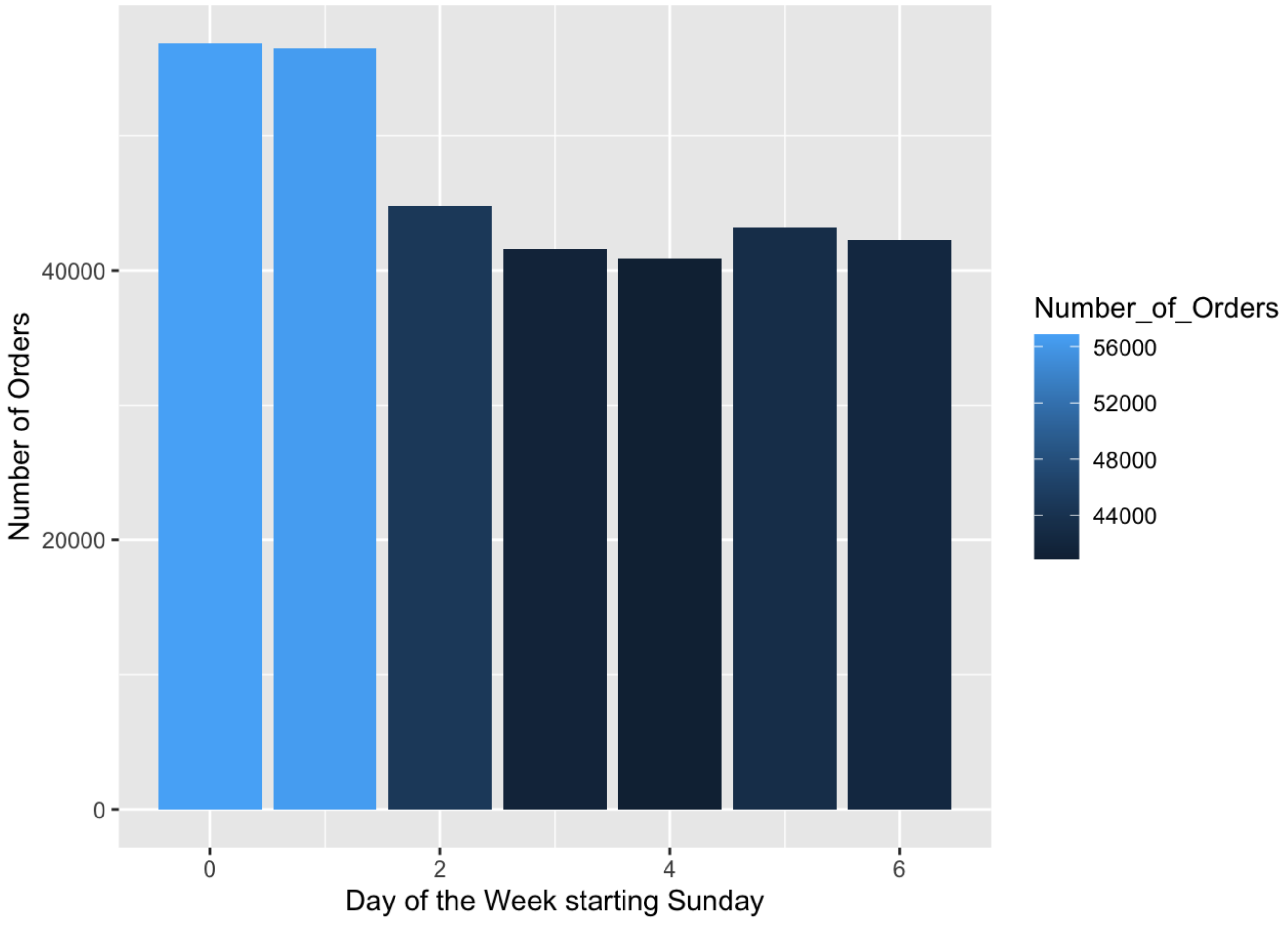
```
##    product_id  product_name      aisle_id department_id
##     "integer"   "character"     "integer"     "integer"
```

DataExploration

1. Ordering patterns

First, we find out most orders are placed during 9am to 5 pm during the day. Additionally, assuming Sunday is the first, ie 0 = Sunday in the dataset, Sundays and Mondays are when most orders are placed.


Number of Orders
20000
10000
0
0
5
10
15
20
25
Hour of the Day
Number_of_Orders
20000
10000

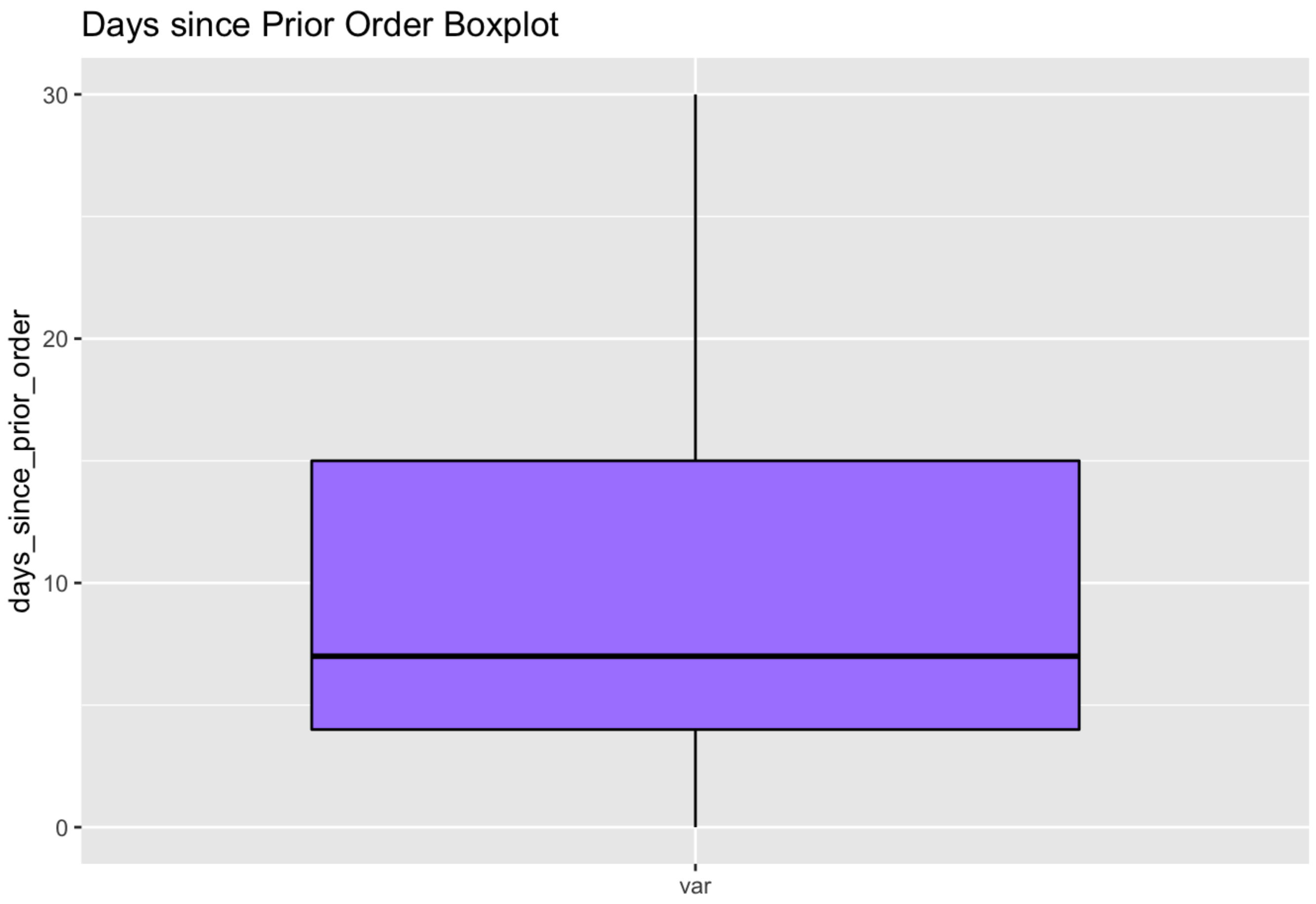

Number of Orders
40000
20000
0
0
2
4
6
Day of the Week starting Sunday
Number_of_Orders
56000
52000
48000
44000

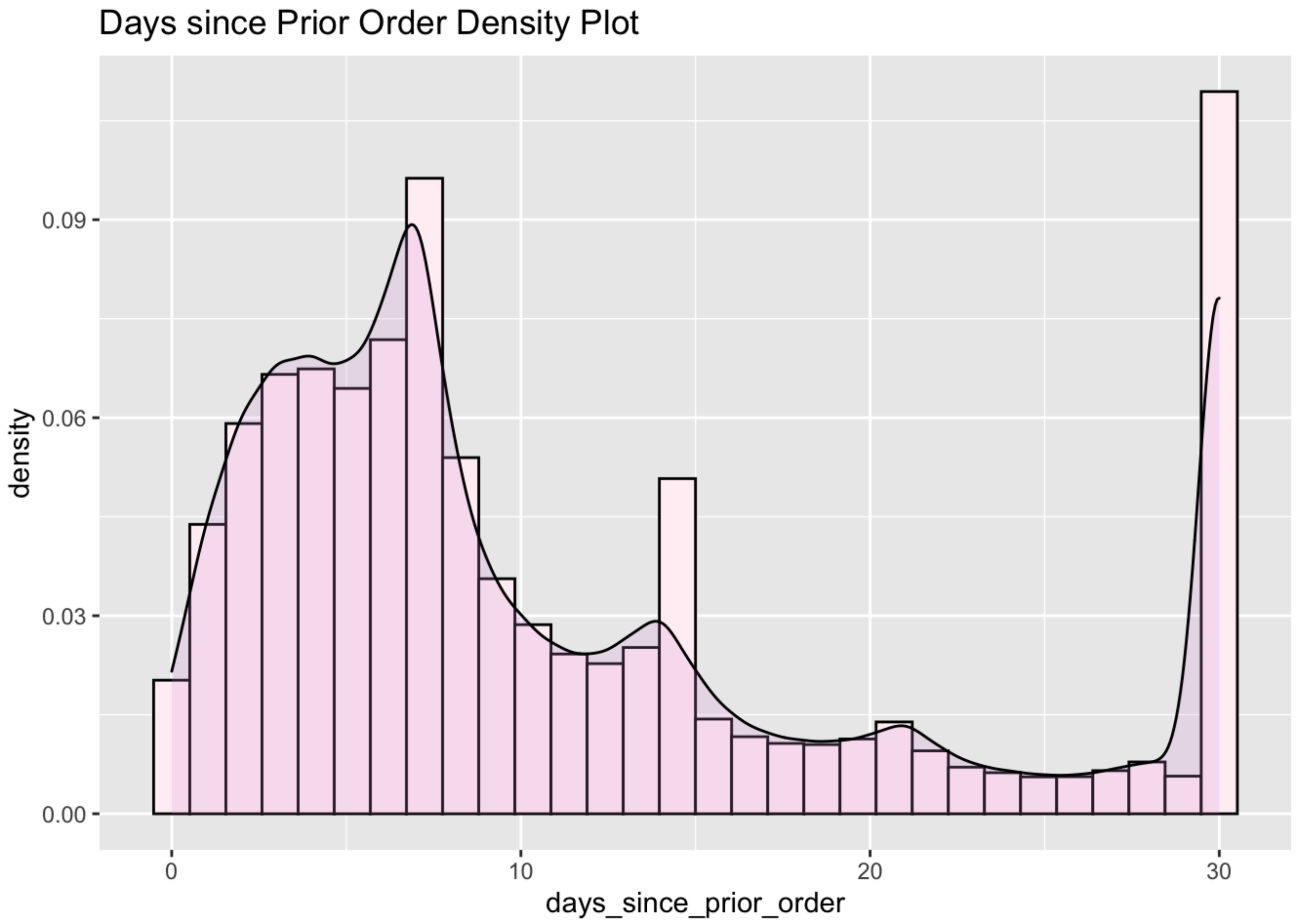
Using this information, Instacart could redirect their resources so that the higher volume of orders can be processed at these times efficiently.

2. What is the frequency of reordering?

When exploring the frequency of reordering, first we filter prior eval_set and select data of days_since_prior_order, then we also clease all the na value from the dataset. We find out in this 3,000,000 orders+ dataset, the average days of users who reorder their groceries is from 7 to 11 days. Many customers reorder after 7 days or 30 days. According to each customer's reordering behavior and how many days they would reorder through instacart,we can predict their next ordering day. Therefore, based on these analysis, it can provide suggestions to retails and suppliers which would be helpful for their purchase sales inventory stategies.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00    4.00    7.00   11.05   15.00   30.00 
```

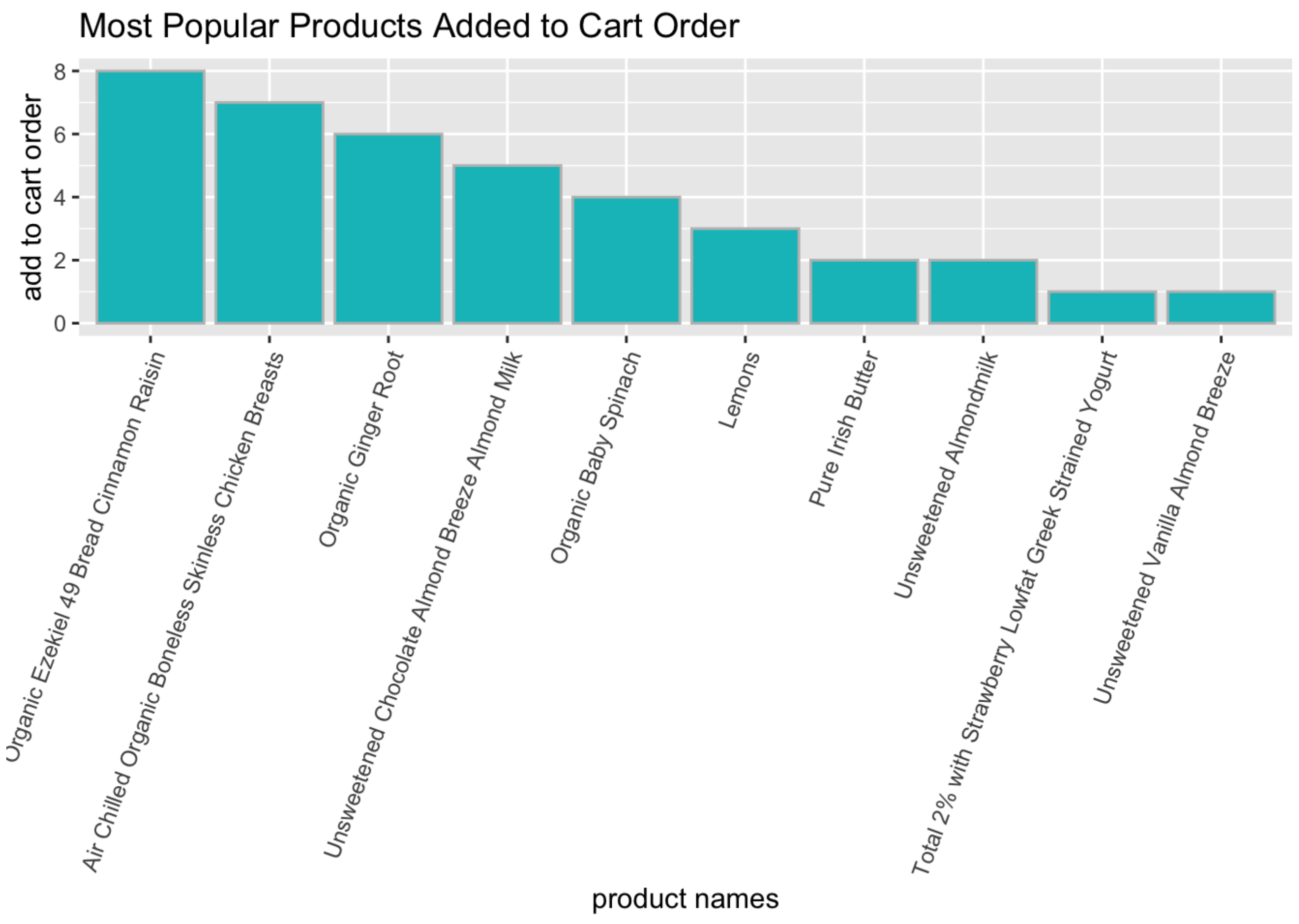

Days since Prior Order Boxplot
days_since_prior_order
30
20
10
0
var


Days since Prior Order Density Plot
density
0.09
0.06
0.03
0.00
0
10
20
30
days_since_prior_order

3. What are the most popular products?

By descending ordering products that are added into cart order, we find out the top 10 popular products out of 49,688 products as shown below. Surprisely, in our analysis, the top No.1 popular product is Organic Ezekiel 49 Bread Cinnamon Raisin.

It is important to find out what products people like to order and order the most, by doing so, we can have better prediction and preparation on the stock of these products. In our next steps, we will find out which departmants and aisles these products belong to, and how often do these top products reorder?

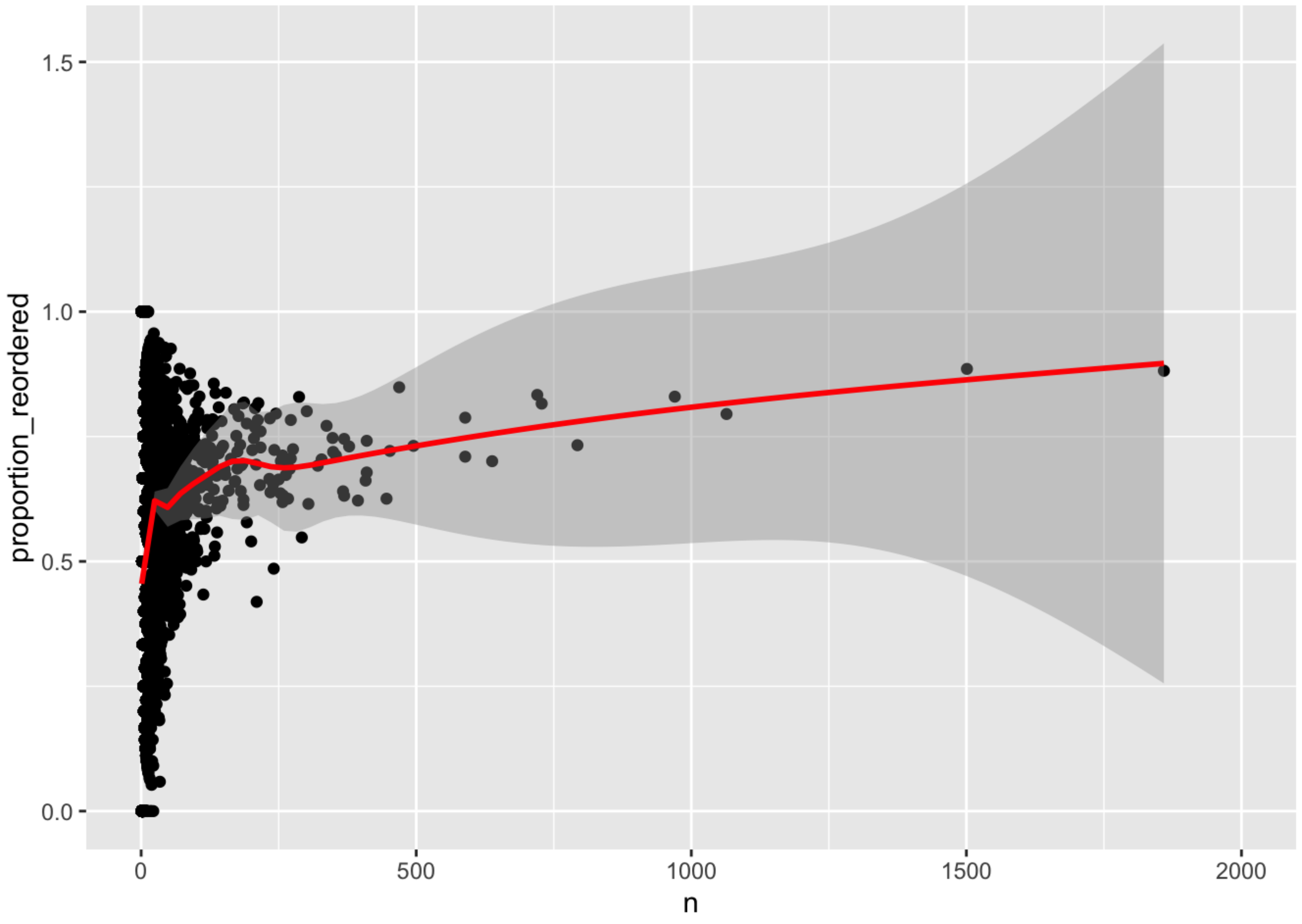


4. Which Users Reorder?

To recommend the next item an user is most likely to add to his/ her cart, we have to understand how each group of users use the service. It becomes a classification problem that we can explore later, but for now we wanted to see how different users purchase products, how there can be correlation between their total number of orders, days since prior order, no. of items they add to their cart and their probability of reordering.

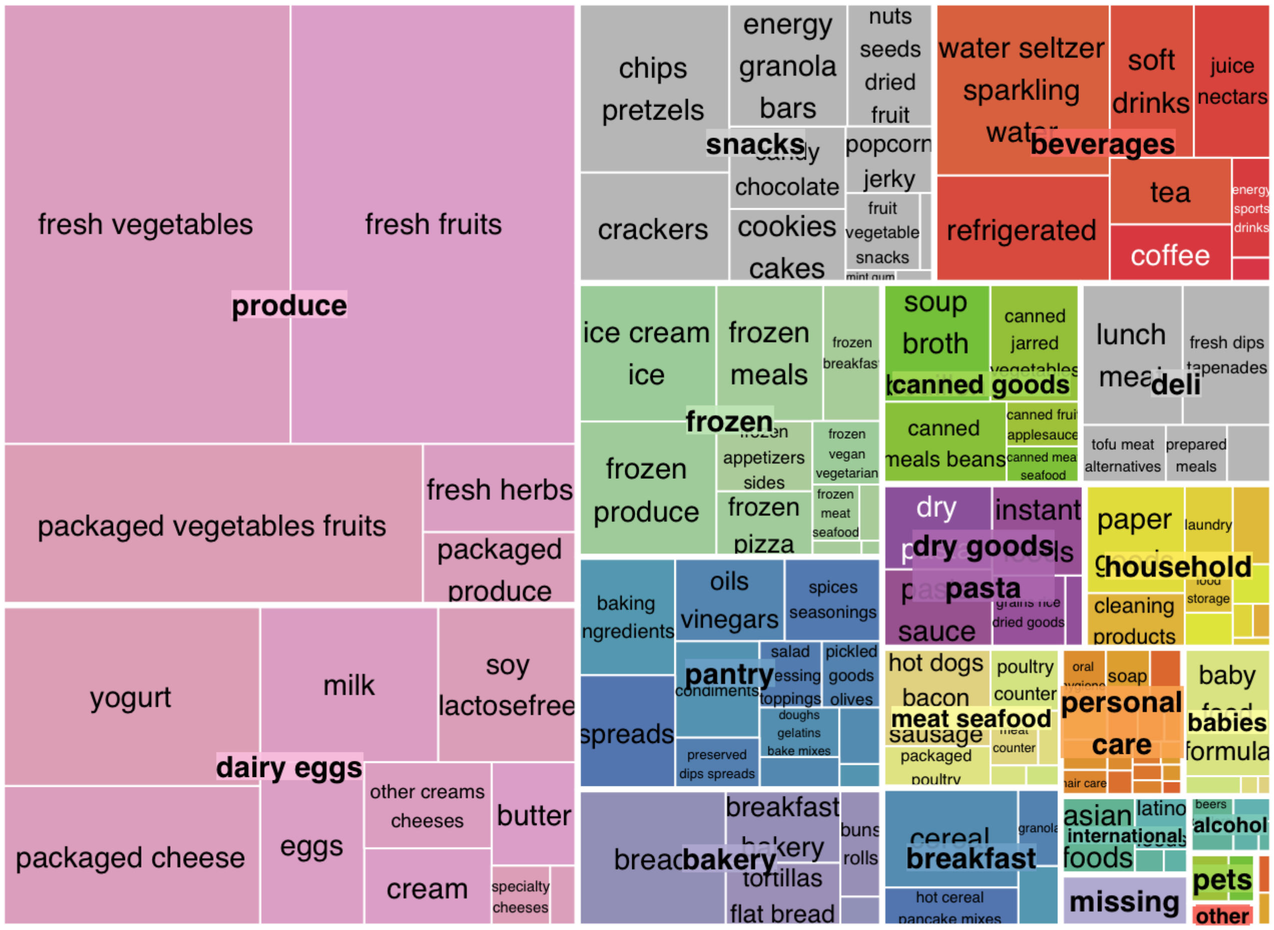
```
## # A tibble: 6 x 5
##   user_id total_orders avg_days_since_prior_order avg_no_items avg_reorder
##     <int>        <int>                      <dbl>        <dbl>       <dbl>
## 1      18            5                       6.47         8.27      0.333
## 2      30            7                      22.6          1.25      0.625
## 3      31           19                       6.07        24.3       0.381
## 4      38           11                      22.9         18.2       0.591
## 5      47            4                       9.75         6.7       0.25
## 6      53            2                      16.5         13.5       0.0370
```

So what we can see from the table the people who have ordered more number of times, they have higher possibility of reordering the same products and they usually order after every three to four weeks. Which means that the products that gets ordered more will also have the higher probablity of being reordered.



5. Most popular department and most popular aisle?

So we wanted to see which are the most popular departments and aisles, as aparently these are the same department and aisles where will most reorder must occur. Later we can also find if which other product has the more probablity of being purchased with other. We can find that using association rule mining. To see the most popular departments and aisles we create a treemap where the size of the boxes shows the number of sales.



What we can see is produce, produce and diary eggs are the most popular departments followed by snacks, pantry and others. But talking about departments, organic and non organic sector divides US consumers in a large way these days. So it will be interesting to see which products get most reordered.

Logistic Regression

We perform logistic regression to predict the variable 'reorder'. Since it is a binary variable with 2 classes,1 and 0, it becomes a classification problem. As we know from our EDA, Instacart has a lot of loyal customers, who order bi-weekly or monthly. While ordering they also reorder a lot of the same products. So predicting whether a user will reorder or not, may help us later to understand what products to recommend while they are purchasing.

```
## # A tibble: 6 x 6
##   total_orders avg_days_since_prior… avg_no_items reordered order_day product_id
##          <int>                 <dbl>        <int>     <int>     <int>      <int>
## 1            5                  6.47           13         1         0      36216
## 2            5                  6.47           13         0         0       4461
## 3            5                  6.47           13         0         0       5876
## 4            5                  6.47           13         0         0        810
## 5            5                  6.47           13         0         0      31717
## 6            5                  6.47           13         1         1      36216
```

So we created a new dataframe dropped the user_id and use all the other columns to predict the reorders with logistic regression. Our regression depends variables such as the total number orders a user has, after how many days he is ordering, how many products he is adding to the cart, which day he is ordering, and which products usually get reordered. For a example if a user usually shops bi weekly, on sunday and orders milk; he might have very less chances of ordering milk if he is back within 2 or 3 days of his last purchase.

Building Model

We use glm with family="binomial" to create the model. We use all of the variables for this model.

```
## 
## Call:
## glm(formula = reordered ~ ., family = "binomial", data = reorder_log)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.2088  -1.2371   0.7585   0.9850   1.4397  
## 
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                9.075e-02  6.503e-03  13.953  < 2e-16 ***
## total_orders               1.500e-02  7.627e-05 196.668  < 2e-16 ***
## avg_days_since_prior_order -2.186e-02  2.955e-04 -73.961  < 2e-16 ***
## avg_no_items               9.700e-03  1.072e-04  90.505  < 2e-16 ***
## order_day                 -1.371e-02  5.839e-04 -23.475  < 2e-16 ***
## product_id                 6.250e-07  8.653e-08   7.222 5.11e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 4025578  on 3061149  degrees of freedom
## Residual deviance: 3854782  on 3061144  degrees of freedom
## AIC: 3854794
## 
## Number of Fisher Scoring iterations: 4
```

Predicting the model

We predict the reorder variable using predict function.

```
## # A tibble: 6 x 6
##   total_orders avg_days_since_prior… avg_no_items reordered order_day product_id
##          <int>                 <dbl>        <int>     <int>     <int>      <int>
## 1            1                     7           11         1         6      36216
## 2            1                     7           11         1         6      47546
## 3            1                     7           11         1         6      21137
## 4            1                     7           11         0         6       5450
## 5            1                     7           11         0         6       8518
## 6            1                     7           11         1         6      22031
```

```
## [1] 0.5351825
```

```
## Accuracy
##     0.54
```

Prediction on the dataset, if p>0.5 then class as 1, otherwise 0 and check the accuracy. As we can see from the result, the prediction of the logistic model is 54% accuracy, which is okay. But it doesn't reveal much information about how well the model actually did in predicting the 1's and 0's independently.

The Confusion Matrix

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction     0     1
##          0 38197 47201
##          1 16611 35275
##                                           
##                Accuracy : 0.5352          
##                  95% CI : (0.5325, 0.5378)
##     No Information Rate : 0.6008          
##     P-Value [Acc > NIR] : 1               
##                                           
##                   Kappa : 0.114           
##                                           
##  Mcnemar's Test P-Value : <2e-16          
##                                           
##             Sensitivity : 0.6969          
##             Specificity : 0.4277          
##          Pos Pred Value : 0.4473          
##          Neg Pred Value : 0.6799          
##              Prevalence : 0.3992          
##          Detection Rate : 0.2782          
##    Detection Prevalence : 0.6221          
##       Balanced Accuracy : 0.5623          
##                                           
##        'Positive' Class : 0               
## 
```

Sensitivity is the percentage of actual 1's that were correctly predicted. It shows what percentage of 1's was covered by the model. The sensitivity is 69.69%, which is okay. Likewise, Specificity is the proportion of actual 0's that were correctly predicted. In this case, it is 42.77%, which does not perform well.

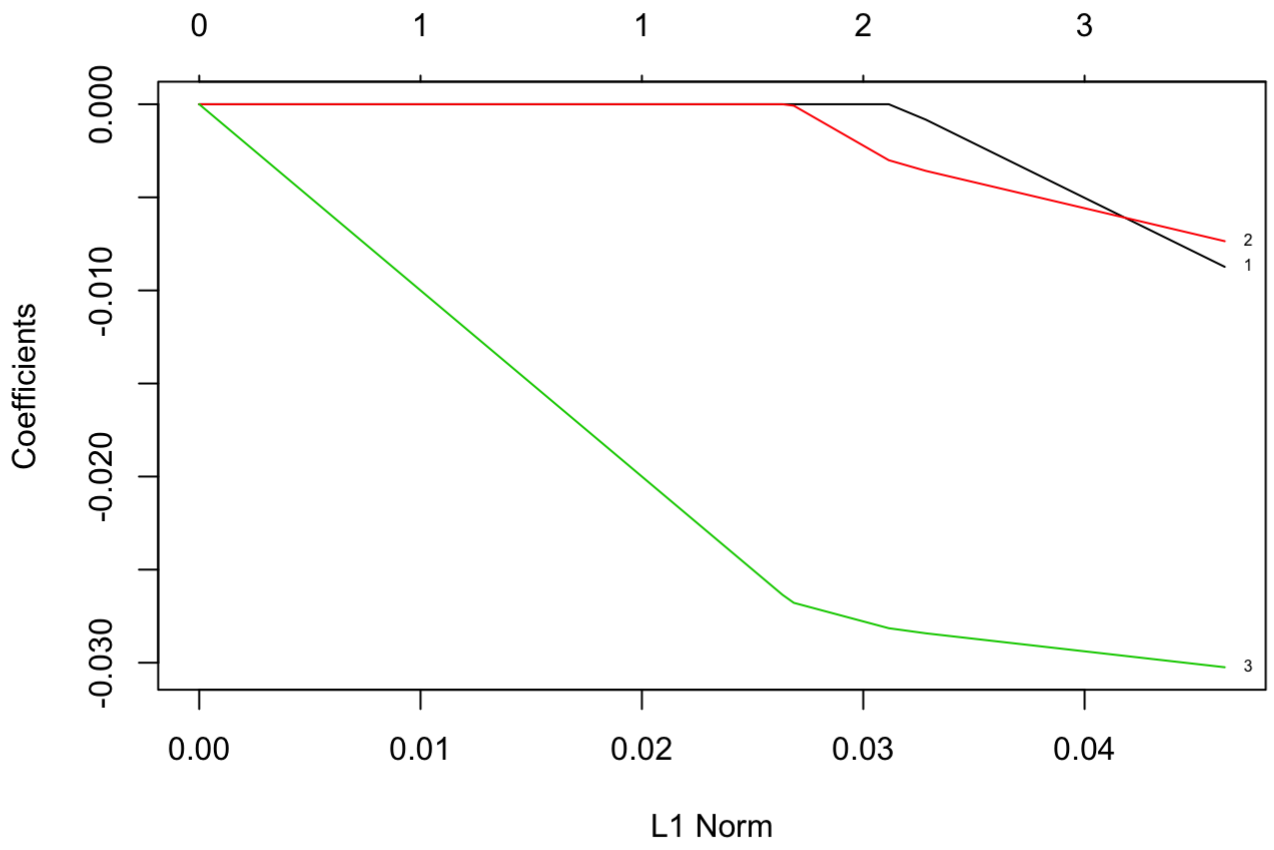
Lasso Regression for regularization

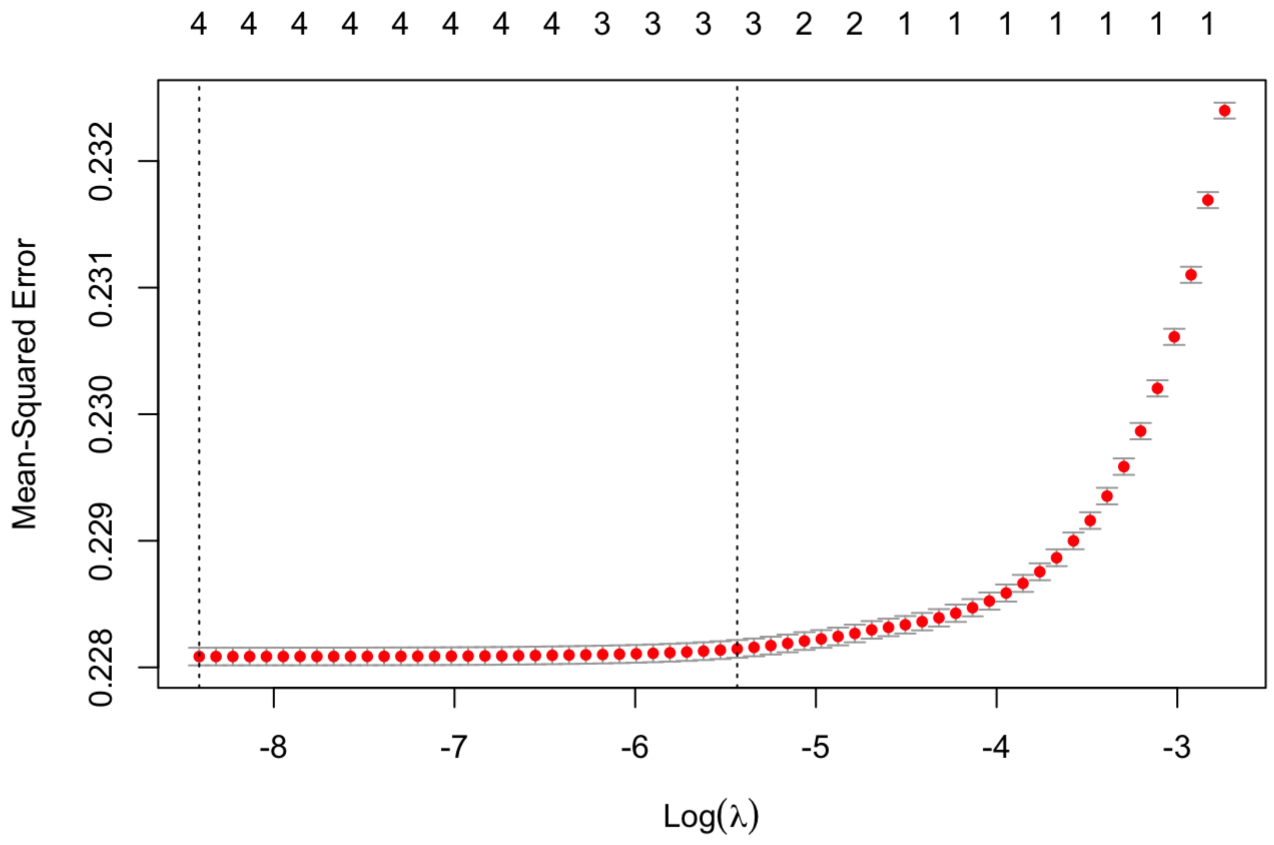
We use lasso regression to remove extra predictors and push them to 0,to try to improve our model.

Lasso plot and ROC

Each colored line in the lasso plot represents the value taken by a different coefficient in the model. Lambda is the weight given to the regularization term (the L1 norm), so as lambda approaches zero, the loss function of your model approaches the OLS loss function (David Marx,2013).

The second plot is the ROC. It shows the lambda values and their MSE. λmin and $$1se are both shown by the vertical line.


0
1
1
2
3
Coefficients
0.000
-0.010
-0.020
-0.030
2
1
3
0.00
0.01
0.02
0.03
0.04
L1 Norm


4 4 4 4 4 4 4 4 3 3 3 3 2 2 1 1 1 1 1 1 1
Mean-Squared Error
0.228
0.229
0.230
0.231
0.232
-8
-7
-6
-5
-4
-3
Log(λ)

```
## [1] "Minimum value of lambda that minimizes mean CV error:  0.0002220"
```

Get estimated beta matrix

Using the min lambda value found, we find the estimated beta matrix. This shows which coefficients have been shrunk to zero and which still exist. This shows which are the important variables that explain the variation in the dependent variable y.

```
## 4 x 1 sparse Matrix of class "dgCMatrix"
##                                  s0
## order_dow              -2.932574e-03
## order_hour_of_day      -2.123839e-03
## days_since_prior_order -7.388845e-03
## product_id              1.116625e-07
```

```
##   alpha lambda
## 3     1  0.002
```

```
## [1] 0.6394443
```

```
## [1] 0.004358156
```

```
## 4 x 1 sparse Matrix of class "dgCMatrix"
##                                  s0
## order_dow              -0.0009051946
## order_hour_of_day      -0.0011581427
## days_since_prior_order -0.0069033275
## product_id              .
```

More variables are removed upon using the lambda 1 standard deviation away. The 1 se rule is a standard one when performing lasso regression. The main point of the 1 SE rule is to choose the simplest model whose accuracy is comparable with the best model, according to (Friedman, Hastie, and Tibshirani,2010).