Optimizing Promotions with Strategic Customer Segmentation & Selection

Kejin Qian, Naomi Kaduwela, Joe Kupresanin, Ruixiang Fan

Executive Summary:

A combined logistic and linear model was built model was built to predict if and how much a customer would purchase during a promotional period for an online bookstore. It is able to predict 27.64% of the actual Euro spent by our top 500 customers, and can recommend a promotional campaign of direct mail to 839 customers will produce a short term profit of €1490.59.

- Key predictors of classification on responders vs non-responders
    - **Tof:** customer time on file, number of days since first purchase
    - **LogMean**: log of average number of books purchased per order
    - **LogSum**: log of the sum of books purchased per order
    - **Rectof:** Rectof is defined as $recency^2/\ tof$
    - **RecLevel * Freq:** Recency is the number of days since the last order, which we bucketed into four levels. Freq represents the $frequency/tof$ on file for that customer. This predictor reveals customer's purchasing habit from both recency and frequency dimensions
- Key predictors for sales in Euros during 3-month promotion period
    - **Last.half.year**: indicates if the customer placed an order in the last 180 days
    - **Mean.items**: the average number of items the customer purchased per order
    - **Median.items**: the median number of items the customer purchased per order

- **Median.total.per.order**: the median amount that the customer spent per order

Introduction:

A German online bookstore has been taking orders since April 1, 2008. Since then, data has been collected on 33,709 customers who have created accounts, many who have purchased items including books and dvds.

A promotion period beginning August 1, 2014 and lasting for 90 days has been implemented by management and our predictive analytics team has been charged with exploring this historical data. The data has been split into a training set of n = 8,311 customers and a testing set of n = 25,398 customers. Using the training data, the team has created two predictive models to strategically target customers to maximize revenue. Since the original data assets included a limited number of variables, our team has developed a number of potential features that are explored in the main discussion.

We assumed that customers who bought frequently would be more likely to buy, unless they just bought recently. We also hypothesized that a customer with the same level of recency, a customer with higher **Freq** is more likely to buy books during the promotion. Also, with the same level of **Freq**, we assumed that the higher the RecLevel(the longer the time from customer's last order), the more likely this person will buy during the promotion. Finally, we assumed that as the sum of a customer's

purchase price increases, so would their likelihood of another purchase. All of these assumptions proved true and thus were translated into features for both models.

First, a logistic regression model was fit to predict whether or not a customer will make a purchase during the promotional period. Though there were challenges like having very few responders in the historical data, in the end, predictors such as time on file, average amount of items bought per order historically, total amount of items bought historically and an interaction term between customer's recency and frequency combined with replicated samples to balance the data resulted in a successful model.

Next, a multiple regression model was developed to predict the amount in Euros a customer will spend during the promotional period. Our final multiple regression model has reasonable predictive power with an adjusted R-sq = 0.3261 including predictors such as the average items per historical order and whether or not a customer has made a purchase in the previous half-year.

The final step was to bring the models together to compute an expected amount purchased by the customers during the promotional period. Using the provided suggestions by management, we found our model to predict 27.64% of the actual Euro spent by our top 500 customers, and can recommend a promotional campaign of direct mail to 839 customers will produce a short term profit of €1490.59.

With reasonable confidence, we can suggest that using our models for future promotional periods can help the company predict incremental sales while suggesting a targeted number of mailings to our customers.

Data Cleaning and Exploratory Data Analysis:

The distribution of the potential variables were examined and skewness was addressed. For brevity, dotplots of all potential features are not shown, but below, clearly the **mean.items**, **median.total.per.order**, and **mean.total.per.order** predictors should be transformed. Logarithms of all variables except **time.on.file** were taken to help distribution symmetry and reduce the impact of outliers.

We removed outliers that had more than 1000 books per order from our training and test sets, as these were rare large purchases and would influence the model. As we can see below in **Figure 0**, most of the original variables and most of the created features are skewed, so we used log transformations in many of our features, e.g. **LogMean, LogNumBooks**.

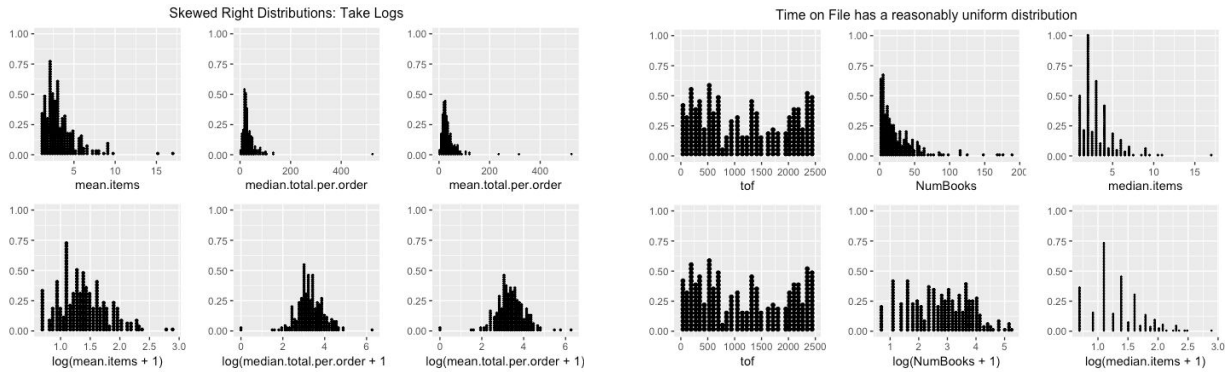


Figure 0: Data Exploration Findings

Logistic Data Preparation

For our logistic model, we noticed that 25% of customers in the training sample bought something for the first time during the test period (previously **frequency** = 0).

Since they have no past purchasing history, all features for a new customer are equal to zero. In this case, if we put both new customers and old customers into the logistic model, all the new customers are more likely to have a lower P(Response) compared to older customers who have nonzero predictors. But in reality, the actual response rate of old customers in the training set is 9.56% while the response rate of new customers in training set is 52.9%, which is much higher. Therefore, we decided to separate all the new customers from the training set before doing logistic regression and just used the existing customers to fit the model. We will use 0.529 as the response rate for all new customers in the test set as the final prediction of their P(response).

To further account for the uneven sample size between responders and nonresponders in the training set, we adopted the strategy of duplicating our sample size twice for the responders to fit logistic regression models. Final predictions on probabilities were then adjusted based on the oversampling.

Logistic Regression Feature Creation

1. **BinaryLogTargAmt:** Binary variable created for the logistic model to determine if customer will purchase:

   if **targAmt** > 0, **BinaryLogTargAmt** = 1; else if **targAmt** = 0,

   BinaryLogTargAmt = 0
2. **Tof:** customer time on file, number of days since first purchase
3. **Mean:** average books per order purchased by customer
4. **NumBooks:** total number of books purchased by each customer over all time

5. **LogMean**: log of average number of books purchased per order
6. **LogSum**: log of the sum of books purchased per order
7. **RecLevel:** bucketing of the **recency** feature provided. Bucket increments: (0, 0.5, 0.75, 0.9, 1). The 0-0.5 bucket is represented by 1, which are customer who have purchased the most recently, whereas the 0.9-1 bucket is represented by 4, which are customers who purchased long time back. We did the bucket increments like this because from **Figure 1** below, we could see that instead of a linear relationship, the relationship between recency and number of customers is roughly exponential. Having this kind of bucket increments can make the recency ranges in the four buckets more similar to each other.

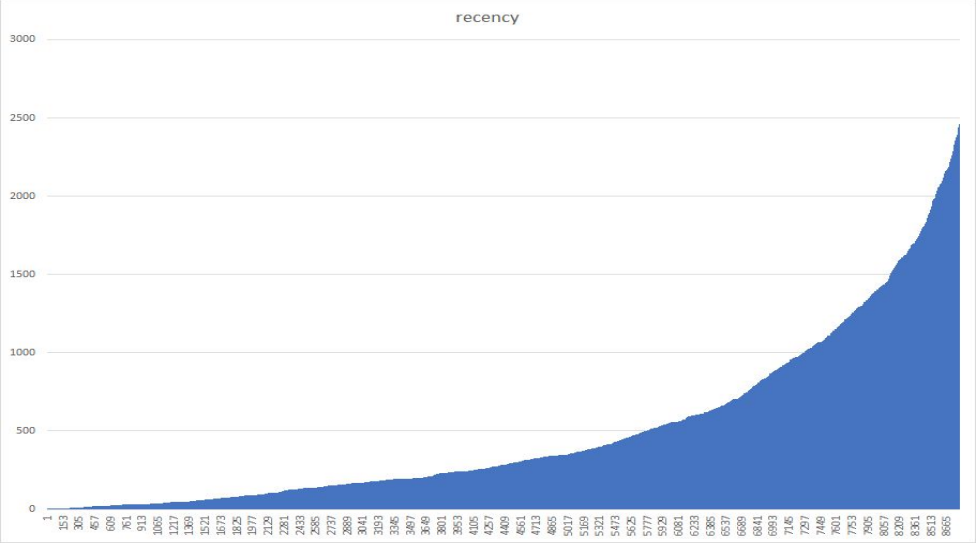


Figure 1: plot of recency to understand bucketing levels

8. **Amount:** total past purchase amount in euros
9. **AmtLevel:** Bucketing **Amount** by quantiles. Bucket increments: (0, 0.25, 0.5, 0.75, 1). The 0-0.25 bucket is represented by 1, which are customer who have purchased the lowest amount before the promotion, whereas the 0.75 - 1 is represented by 4, the customers who purchased the most before the promotion.
10. **Freq**: **frequency** / **tof** where **frequency** is number of orders a customer placed in the past and **tof** is time on file. This predictor stands for average number of orders per one day on file and is used to determine if a customer is a frequent book buyer. We think that **frequency** itself cannot correctly show a customer's purchasing habits. Because Customer A and Customer B can have the same frequency count but A may had these orders in 10 years while B ordered this much in only 6 months. In this case, B actually purchased books much more frequently than A and should be more likely to purchase again during the promotion.
11. **RecLevel * LogSum**: Interaction Term between **RecLevel** and **LogSum**. Recency is the number of days since the last order, which we bucketed above. **LogSum** is the log of **NumBooks**, which represents total number of books purchased by a customer in the past.
12. **RecLevel * Freq**: Interaction Term between **RecLevel** and **Freq**. Recency is the number of days since the last order, which we bucketed above. Freq represents the frequency / time on file for that customer. We thought it would be meaningful to multiply this together to examine our assumption: With the same level of

recency, a customer with higher Freq is more likely to buy books during the promotion, and with the same level of Freq, the higher the RecLevel(the longer the time from customer's last order), the more likely this person will buy during the promotion.

13. **Rectof: Rectof** is defined as (recency^2)/tof. We added this predictor in order to distinguish two kinds of customers. First, those with large **tof** and large **recency** - this group of customers have been inactive for a long time and we think they will be less likely to buy during the promotion. Second, those with small **tof** and small **recency** - this group of customers recently came to the website to place their first order, so they are relatively active customers and will be more likely to be attracted by the promotion. Without the square (simply **recency**/**tof**), customers from both groups have similar numbers, but the numbers differ considerably after squaring **recency**. Also, from the training set, we calculated that the average **Rectof** of all the old customers is 349.0303 and the average **Rectof** of all the responders is 169.1627 which is much smaller compared to the overall average. This indicates that **Rectof** may be a good predictor for determining whether an old customer will purchase during the promotion and it shows that the lower the **Rectof**, the more likely this customer will purchase during the promotion.

Logistic Regression Feature Selection:

Aftering coming up with 13 features we believed would determine if a customer was likely to buy during the promotion or not, we used the MASS library and ran a stepwise regression in both directions to select the best set of predictors (**Figure 2**). From our initial model we saw that 4 predictors were dropped: **mean**, **amount**, **amtLevel** and **LogSum * RecLevel**

```
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
Binarylogtargamt ~ tof + mean + NumBooks + logMean + logSum +
    freq + amount + amtLevel + freq:recLevel_ + logSum:recLevel_ +
    rectof

Final Model:
Binarylogtargamt ~ tof + NumBooks + logMean + logSum + freq +
    rectof + freq:recLevel_


                Step Df     Deviance Resid. Df Resid. Dev      AIC
1                                         8772   5156.378 5180.378
2             - mean  1 9.288406e-05      8773   5156.378 5178.378
3           - amount  1 8.225883e-02      8774   5156.460 5176.460
4         - amtLevel  1 1.300686e+00      8775   5157.761 5175.761
5 - logSum:recLevel_  1 1.649812e+00      8776   5159.411 5175.411
```

Figure 2: Stepwise Regression output to select best predictors

We first fitted a logistic regression on the remaining predictors.

Binarylogtargamt ~ tof + NumBooks + logMean + logSum + freq + rectof + freq:recLevel_

The summary output of this model is given in **Figure 3**:

```
Call:
glm(formula = Binarylogtargamt ~ tof + NumBooks + logMean + logSum +
    freq + rectof + freq:recLevel_, family = binomial, data = data_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1140  -0.4799  -0.3709  -0.2914   2.7714

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    -2.742e+00  1.306e-01 -20.998  < 2e-16 ***
tof            -6.905e-04  7.395e-05  -9.338  < 2e-16 ***
NumBooks       -6.991e-03  2.083e-03  -3.356 0.000792 ***
logMean        -6.380e-01  1.008e-01  -6.329 2.46e-10 ***
logSum          8.983e-01  8.022e-02  11.198  < 2e-16 ***
freq            1.049e+02  1.636e+01   6.415 1.41e-10 ***
rectof          1.826e-04  1.471e-04   1.242 0.214331
freq:recLevel_ -9.519e+01  1.628e+01  -5.848 4.97e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5540.4  on 8783  degrees of freedom
Residual deviance: 5159.4  on 8776  degrees of freedom
AIC: 5175.4

Number of Fisher Scoring iterations: 6
```

Figure 3: Summary output of

Binarylogtargamt ~ tof + NumBooks + logMean + logSum + freq + rectof + freq:recLevel_

To evaluate the current model, we decided to check for multicollinearity and ran the VIF scores among all the 7 predictors we currently have. **Figure 4** shows our findings.

tof	NumBooks	logMean	logSum	freq	rectof	freq:recLevel_
2.558203	2.810430	2.085098	6.319819	67.006038	1.962935	65.042264

Figure 4: VIF Scores to test Multicollinearity

We can see that **freq*recLevel** has a very high VIF 65.042264 which is much larger than 10, so we confirmed that there exists multicollinearity problem in our current

model. To further analyze this multicollearity, we plotted the correlation matrix and calculated the correlations among the existing predictors. Outputs are presented in **Figure 5** and **Table 1**.

	tof	logMean	logSum	NumBooks	rectof	freq	freqrecLevel_
tof	1.00000000	0.05829175	0.51528887	0.3682171	0.21311288	-0.20745480	-0.15096178
logMean	0.05829175	1.00000000	0.59180356	0.4051112	-0.08585822	-0.01621024	-0.01729632
logSum	0.51528887	0.59180356	1.00000000	0.7513829	-0.38769711	0.03464404	0.06256388
NumBooks	0.36821714	0.40511117	0.75138288	1.0000000	-0.26934051	0.10757990	0.12231181
rectof	0.21311288	-0.08585822	-0.38769711	-0.2693405	1.00000000	-0.20169089	-0.14800859
freq	-0.20745480	-0.01621024	0.03464404	0.1075799	-0.20169089	1.00000000	0.97984256
freqrecLevel_	-0.15096178	-0.01729632	0.06256388	0.1223118	-0.14800859	0.97984256	1.00000000

Table 1: Correlation between predictors in model

Binarylogtargamt ~ tof + NumBooks + logMean + logSum + freq + rectof + freq:recLevel_

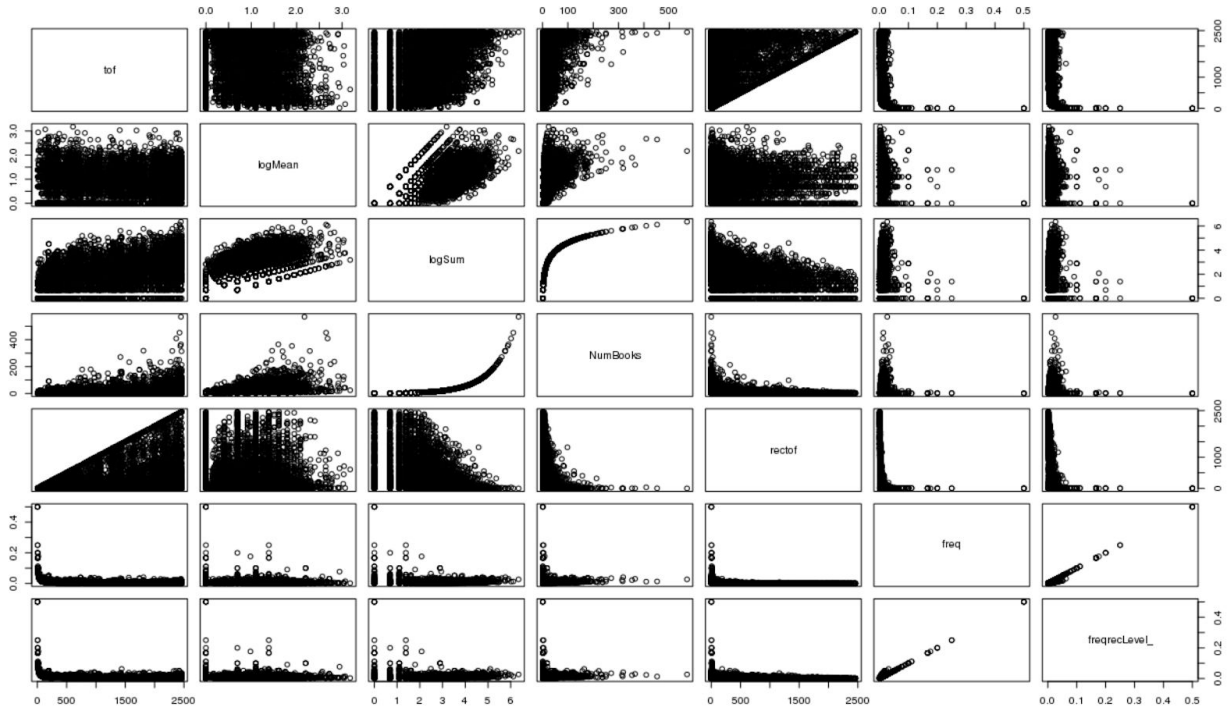


Figure 5: Correlation Matrix of predictors in model

Binarylogtargamt ~ tof + NumBooks + logMean + logSum + freq + rectof + freq:recLevel_

From the correlation matrix and correlation table, we see that **freq** and **freqrecLevel** were highly correlated(0.97984256) and the correlation plot shows a strong, positive linear relationship. We chose to drop **freq** as we thought **freqrecLevel** was more informative for the model since it considered two dimensions of customer buying behavior(frequency and recency).

Next, we also found that **LogSum** and **NumBooks** are also highly correlated (0.7513829), which makes sense since **LogSum** is simply log(**NumBooks**). But we noticed that **LogSum** has a significant positive beta coefficient equals to 8.983e-01 with p-value < 2e-16 while **NumBooks** has a significant negative beta coefficient equals to -6.991e-03 with p-value 0.000792. This looks strange to us, since our **LogSum** is simply the log of **NumBooks**, thus these number should increase and decrease simultaneously. So we did not expect the beta coefficients to have opposite signs. In terms of model interpretation, it doesn't make sense to have **LogSum** and **NumBooks** with opposite signed coefficients, so we decided to drop one of them out of the model. Since **LogSum** had a much more significant p-value and larger absolute magnitude of beta coefficient, we decided to keep **LogSum** and drop **NumBooks**. Also from general intuition, we think a customer who bought more books in the past will be more likely to buy during the promotion, so it's better to keep the predictor with positive beta coefficient. Apart from these two pairs of predictors (**freq** & **freqrecLevel**, **LogSum** & **NumBooks**), we didn't see any problematic correlations and the scatter plots did not show any clear linear relationships.

At this point, we reran the model with the remaining predictors from the above the selection process. After implementing the changes, our current model is:

Binarylogtargamt ~ tof + logMean + logSum + rectof + freq:recLevel_

The model summary is presented in **Figure 6:**

```
Call:
glm(formula = Binarylogtargamt ~ tof + logMean + logSum + rectof +
    freq:recLevel_, family = binomial, data = data_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6801  -0.4932  -0.3939  -0.2978   2.9968

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -2.460e+00  1.078e-01 -22.828  < 2e-16 ***
tof            -7.076e-04  7.393e-05  -9.572  < 2e-16 ***
logMean        -5.388e-01  9.993e-02  -5.392 6.97e-08 ***
logSum          6.681e-01  6.283e-02  10.634  < 2e-16 ***
rectof         -2.918e-04  1.471e-04  -1.984   0.0473 *
freq:recLevel_  7.189e+00  1.801e+00   3.991 6.58e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5540.4  on 8783  degrees of freedom
Residual deviance: 5217.3  on 8778  degrees of freedom
AIC: 5229.3

Number of Fisher Scoring iterations: 6
```

Figure 6: Summary output of

Binarylogtargamt ~ tof + logMean + logSum + rectof + freq:recLevel_

From the summary output, we can see that all the 5 predictors in our current model are significant, and **tof**, **logMean**, **logSum** have very small p-values(<e-08). In order to make sure the multicollinearity problem has been completely eliminated by the

previous prediction selection procedure, we calculated the VIF scores of all the predictors again.

tof	logMean	logSum	rectof	freq:recLevel_
2.598358	2.163344	4.258005	1.538658	1.055300

We can see that all the VIFs are very small (the largest one is VIF(logSum) = 4.258005 < 10).

Logistic Regression Model Interpretation

Based on the analysis above, our final logistic regression model is

$$log(\frac{p(respond)}{1-p(respond)}) = -2.46 - 7.076 \times 10^{-4} \cdot tof - 0.5388 \cdot logMean + 0.6681 \cdot logSum - 2.918 \times 10^{-4} \cdot rectof + 7.189 \cdot freq : recLevel$$

Coefficient Interpretation

1. **tof** : the predictor tof has a negative β coefficient of -7.076×10^{-4} . This shows that by increasing time on file by one day, the log odds of being an responder vs. non-responder will decrease by 7.076×10^{-4} . In other words, the longer the time a customer stays on file, the less likely this customer will buy during the promotion.
2. **logMean**: the predictor logMean has a negative β coefficient of -0.5388. It implies that 1% increase in log of average amount of books bought per order will decrease the log odds of being an responder vs. non-responder by 0.5388%.
3. **logSum**: the predictor logSum has a positive β coefficient of 0.6681. It means that 1% increase in the log of total amount of books a customer ever bought

leads to a 0.6681% increase in the log odds of being an responder vs. non-responder.

4. **rectof**: the predictor **rectof** has a negative β coefficient of -2.918×10^{-4}. This shows that one-unit increase in **(recency^2)/tof** will decrease the log odds of being an responder vs. non-responder will decrease by -2.918×10^{-4}. This confirmed our assumption we made when we first created this feature: The lower the **rectof**, the more likely this customer will purchase books during the promotion.
5. **freq:recLevel**: the predictor **freq:recLevel** has a positive β coefficient of 7.189, which shows that one-unit increase in **freq*recLevel** will increase the log odds of being an responder vs. non-responder by 7.189. Again, this confirms our initial assumption that with the same level of **freq**, customers with higher **recency** are more likely to respond during the promotion and for customers with the same **recLevel** (bucketed recency), the customers who bought books more frequently will be much more likely to purchase during promotion.

Since we adopted the strategy of oversampling the responders in fitting logistic regression models, when we were making final predictions, we did a transformation on the probabilities based on the following formula:

$$\log[p1/(1-p1)] = -\ln(2) + \ln[p2/(1-p2)]$$

where p1 denotes the fitted probability with no oversampling and p2 denotes the fitted probability with 2-fold oversampling.

Linear Regression Feature Creation

- **Average.time.between.order**: This variable measures the average time between one customer's orders. We created it because if customers had regular purchases in last few years, e.g. every 3 months, then it would be highly possible that they would place orders in the 3-month promotion period.
- **Average.amount.spent.over.time**: This variable measures the average amount one customer spent in a certain length of time, e.g. 3 months. We created it because if customers spent regular amount in bookstore every 3 months, then it would be a pivotal part of their spent in promotion period.
- **Last.30.days**, **Last.60.days**, … , **Last.360.days**: These variables indicate if the customer made orders in certain month within last year. For example, **Last.30.days** represented last month, **Last.60.days** represented the month before last month, etc. We created these variables because we could find if the customer made purchases in a regular basis. Besides, if the customer made purchases most recently, they may not make more purchases as they needed time to finish reading. And also, if the customer made regular purchases a while ago but none recently, it may indicate that they switched to other stores, meaning it would be less likely for them to make any orders in promotion period.

Linear Regression Feature Selection

The multiple regression model incorporates data from the book and the orders datasets. Below are the most pertinent features that were created, although not all of them ended up in the final model. In addition, existing predictor variables were considered in the analysis.

1. **median.items & mean.items:** Per customer id, aggregated to per-order values
2. **median.total.per.order & mean.total.per.order:** Euros spent per customer id, per order
3. **days.since.last.order:** Computed from last order date in the system
4. **median.per.item.spent & mean.per.item.spent:** Euro averages per customer id per item on orders
5. **last.30:** Indicator variable if customer made purchase in last 30 days
6. **last.60:** Indicator variable if customer made purchase exactly 31 - 60 days ago
7. **last.90, last.120, ... , last.360:** Same idea as **last.60**
8. **last.half.year:** Indicator variable if customer made purchase within last 180 days
9. **avg.time.between.orders:** Measured in days. If customer ordered exactly one time, we replaced with time since last order.
10. **one.timer.purchaser:** Indicator if customer has exactly one purchase in history
11. **freq.tof:** Frequency of purchase / time on file. Larger number indicate more purchases more recently.
12. **avg.time.between.orders * recency:** Interaction term

Feature Cleaning

We found there were 46 new customers out of 327 responders after the promotional offer was made. Owing to its significant percentage of the whole group, we decided that they cannot simply be dropped. Thus, different models were built for old customers and new customers. A multiple linear regression model was built on 281 existing customers in training set. While there was no historical data available for new customers (the only information we had was their purchase amount after the promotion offer was made), we decided to use the average amount spent as the prediction for new customers. To separate these two groups of customers, we explored if the customer had prior orders. A column named new.Customer was added, and we coded new customer with 1, and existing customers with 0. We then split the responders by the new.Customer column into two groups.

A Lesson Learned (Infinite Time Between Orders):

Earlier in the project work, as multiple regression models were being fit, we were getting better performance using measures of fit such as an adjusted R-sq = 0.3625 and an RSS = 0.5737. One variable that kept showing up in our model was **avg.time.between.orders**, but an erroneous decision had been made to drop rows where **avg.time.between.orders** = Inf. As it was reasoned that these were customers

who had purchased exactly one time prior, we made the decision to replace the Inf values with **days.since.last.order** and then create an indicator to flag customers who were **one.time.purchasers**. Model measures of fit decreased, but this decision was the correct approach to address our error.

Weighted Indicators:

One variable not mentioned thus far was a weighted linear combination of

$$6.6(\text{last.30}) + 6.3(\text{last.60}) + \ldots + 4.0(\text{last.330}) + 4.0(\text{last.360})$$

The initial thought was that since more recent purchases might indicate a customer was more likely to buy during the promotion period, we could weight the indicators with emphasis on the near term. During the model fitting, only one of these indicators was significant individually (the weighted linear combination was not, and various decreasing weight schemes were evaluated using trial and error).

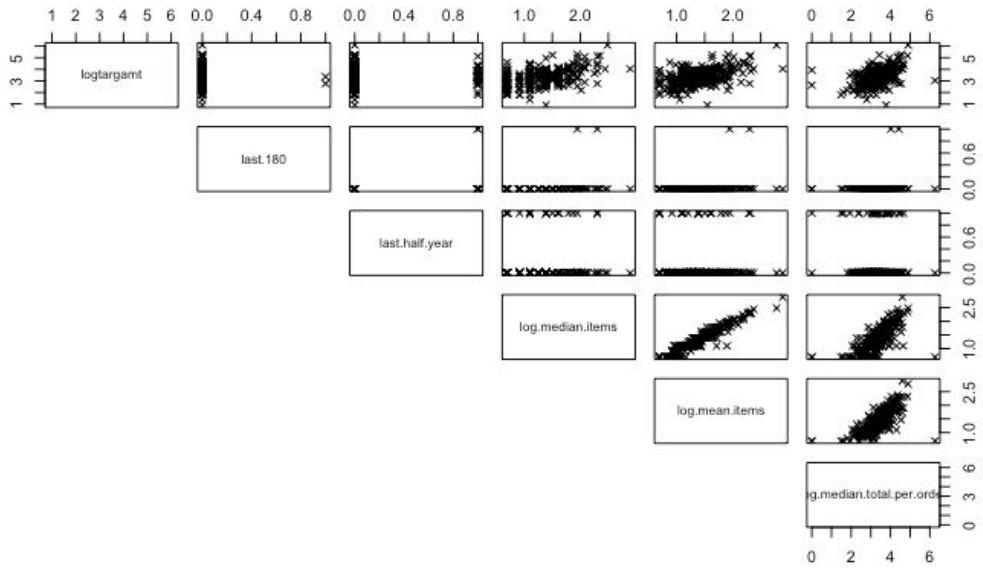
Feature Selection:

For the multiple linear regression portion, we chose to use the MASS package, which performs stepwise regression in both directions. For variable selection, the stepAIC algorithm was chosen, which only includes complete cases. Initially, all potential predictor variables are included in the full model before implementing the algorithm to reduce the number of features.

```
Call:
lm(formula = logtargamt ~ last.180 + last.half.year + log.mean.items +
    log.median.items + log.median.total.per.order, data = book.train.oldCustomer)

Residuals:
     Min       1Q   Median       3Q      Max
-2.66353 -0.40806  0.02107  0.41080  1.84191

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                 1.59825    0.18487   8.646 4.57e-16 ***
last.180                   -1.15695    0.45035  -2.569 0.010728 *
last.half.year              0.23205    0.12610   1.840 0.066832 .
log.mean.items              1.25160    0.32925   3.801 0.000177 ***
log.median.items           -0.54796    0.29404  -1.864 0.063457 .
log.median.total.per.order  0.21250    0.07718   2.753 0.006292 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6048 on 274 degrees of freedom
Multiple R-squared:  0.302,     Adjusted R-squared:  0.2893
F-statistic: 23.71 on 5 and 274 DF,  p-value: < 2.2e-16
```

At this point, condition checking and outlier detection began. Concern exists for including both **log.mean.items** and **log.median.items** in the model, especially with opposite coefficient signs. Averaging the two together was explored, but model selection always includes both predictors in the final equation based on minimizing AIC.

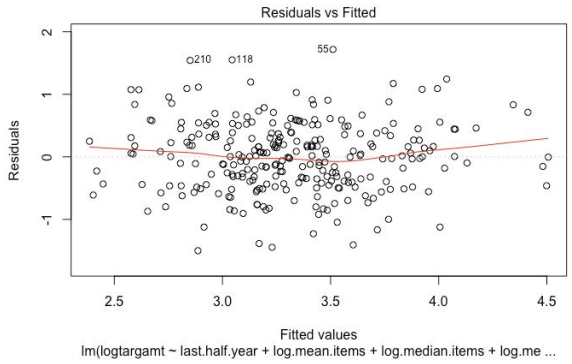
Three observations were removed on account of visual inspection - the top right display shows two observations with a **log.median.total.per.order** = €0 and one with €6.2628.

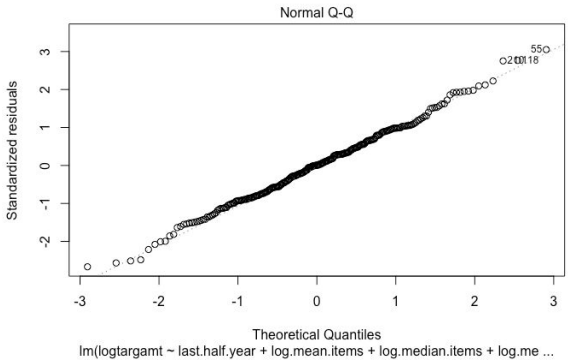


After refitting, we had three observations with Cook's distances of 0.2106, 0.2106, and 0.0627 (cutoff of 0.0433) and one observation with a standardized residual of -4.605. On account of simply improving model fit, we removed these four observations and refit the model. Because our indicator variable **last.180** only had two observations coded at 1, this predictor was unintentionally dropped from the model at this point.

```
Call:
lm(formula = logtargamt ~ last.half.year + log.mean.items + log.median.items +
    log.median.total.per.order, data = book.train.oldCustomer)

Residuals:
     Min       1Q   Median       3Q      Max
-1.50042 -0.40495  0.00495  0.37014  1.71710

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                1.23914    0.20522   6.038 5.18e-09 ***
last.half.year             0.21380    0.11983   1.784  0.07552 .
log.mean.items             0.97610    0.32560   2.998  0.00297 **
log.median.items          -0.53540    0.27819  -1.925  0.05534 .
log.median.total.per.order  0.43146    0.09576   4.506 9.89e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5652 on 268 degrees of freedom
Multiple R-squared:  0.336,     Adjusted R-squared:  0.3261
F-statistic:  33.9 on 4 and 268 DF,  p-value: < 2.2e-16
```

No model is perfect, and this multiple regression has its deficiencies. In particular, we have the issue of both **log.mean.items** and **log.median.items** being included, and if one or both are removed, penalties exist with respect to model fit and the significance of the other predictors. As these two variables are logically related (also see the scatterplot), we have VIFs of 13.205 and 11.247 respectively. While exceeding the threshold of 10, we decided to leave the model as is to retain its predictive capability. As evidenced below, we feel the residuals are homoscedastic with respect to the predicted **logtargamt** and our residuals seem to be reasonably normally distributed.





Multiple Regression Model Interpretation

Based on the analysis above, our final multiple regression model is

$$logtargamt = 1.23914 + 0.2138 \cdot last.half.year + 0.9761 \cdot log.mean.items - 0.5354 \cdot log.median.items + 0.4315 \cdot log.median.total.per.order$$

last.half.year = 0.2138: If a customer has purchased in the immediate 180 days before the promo period, our predicted **logtargamt** increases by 0.2138, which is an expected increase of 0.238 Euros spent during the promo period.

log.mean.items = 0.9761: For previous purchasers, if the mean items per order increased by 1%, we expect target amount to increase 0.9761%

log.median.items = -0.5354: Keeping in mind the above interpretation, this predictor seems to be counteracting the increase due to **log.mean.items** being positive. Perhaps a customer with many previous purchases had a few many-item purchases that would contribute to a higher mean - this media-based term is reducing its impact on promo predicted amount. If the median items per order increased by 1%, our model predicts promo period spending to decrease by 0.5354%

log.median.total.per.order = 0.4315: For previous purchasers, if the median total per order increases by 1%, our model predicts promo spending to increase by 0.4315%.

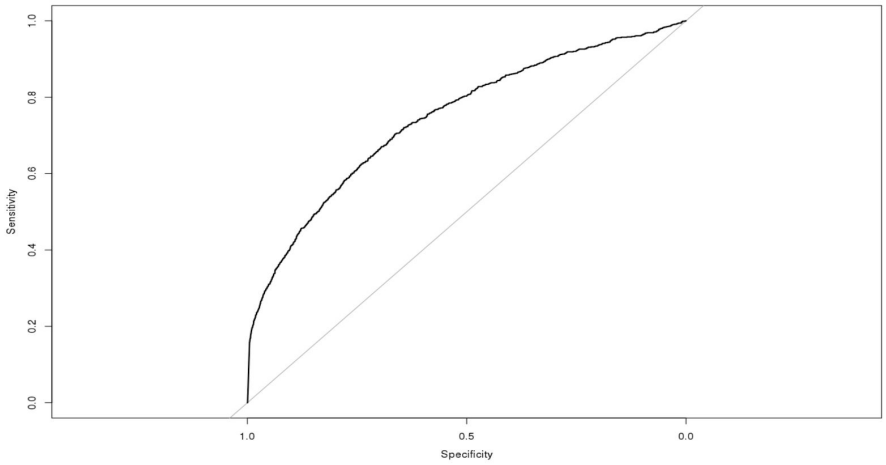
Logistic Model Validation

$$log(\frac{p(respond)}{1-p(respond)}) = -2.46 - 7.076 \times 10^{-4} \cdot tof - 0.5388 \cdot logMean + 0.6681 \cdot logSum - 2.918 \times 10^{-4} \cdot rectof + 7.189 \cdot freq : recLevel$$

Our final logistic model attained an AIC of 5229.3 which is a little higher than the minimum AIC we could get from the final stepAIC model (AIC = 5175.441). However, compared to the final stepAIC model, our current model has all predictors highly significant and also eliminates the problem of multicollinearity.

As we did in the training set, we split our test set into new and old customer groups. The response rate of all the new customers are predicted as 0.529 which is the response rate of new customers in the training set. The response rate of all the old customers is predicted by our final logistic regression model and we applied ROC to evaluate the classification model performance. The AUC of our model on the test set is 0.7007 which is clearly larger than 0.5. The AUC plot is presented below.

AIC: 5229.3
AUC: 0.7007



AUC curve for Logistic Regression = 0.7007

Multiple Regression Model Validation

As what we did in train set, we split test set into old and new customer groups. The amount that old customers spent after promotional offer was predicted by the multiple linear regression model we built, while the average amount by new customers in training set was used here to predict new customers in test set.

A full model could be fit with $p = 29$ predictors, but this falls victim to overfitting without much added predictive capability. For comparison's sake, predicted **logtargamt** based on all 29 features had:

RSS: 0.5795 on 244 degrees of freedom with an overall P-value = 1.017e-12

Multiple R-squared: 0.3645

Adjusted R-squared: 0.2916

AIC: 506.1739

BIC: 614.4581

A forward / backward algorithm that minimizes AIC was fit using the MASS package and the stepAIC function in R. The algorithm reduced the number of predictors down to 5, and removal of the extremely unusual data points led to the removal of another predictor variable (**last.180**).

RSS: 0.5652 on 268 degrees of freedom with an overall P-value of 2.2e-16

Multiple R-squared: 0.336

Adjusted R-squared: 0.3261

AIC: 470.1754

BIC: 491.8322

After we combined the classification model and multiple linear regression model, for each observation (including both new and old customers) in the test set, we calculated E(**logtargamt**) by multiplying the predicted **logtargamt** from the multiple regression model by P(**logtargamt** > 0) from the logistic regression model by using the formula E(y) = E(y|y > 0)P(y > 0). This gives the predicted **logtargamt** for the test set customers. We validated our final model based on both statistical and financial criterion.

Statistical Criterion

The sum of squared errors of prediction (SSEP) is obtained by summing the squares of the differences between the actual **targamt** and the predicted **targamt** values for the test sample.

SSEP: 2,471,274

Financial Criterion

The top 500 customers from the test set who have the highest E(targamt) have actual total purchases of €7472.27, which is 27.63916% of the total purchases of the actual top 500 customers in the test sample.

Given the profit margin is 25% of targamt and the cost of mailing the promotional material to each prospect is 1 euro, the optimum number of top prospects we should mail the promotional material to is 839 and will gain a short term profit of €1490.586.

% in Top 500: 27.63916%

Optimum Payoff: €1490.586

Number of Customer we need to mail promotions to get payoff: 839

Conclusion

This combined logistic and linear regression model will indeed prove useful during future promotional periods, as it can enable efficiencies in customer targeting and drive a short term profit of €1490.586 by predicting 27.64% of the actual Euro spent by the top 500 customers and recommending direct mailings to 839 customers.

By employing statistically strategic techniques, many challenges with the data were overcome, such as replicating samples to address an unbalanced dataset and leveraging the data we did have to populate predictions for customers that did not have historical data.

Our initial hypotheses that when when a customer made a purchase and how much they spend would be key predictors in our model proved true. Features were built upon on those underlying assumptions, such as time on file, average amount spent

historically, average items per historical order and whether or not a customer has made a purchase in the previous half-year.

In the future, the model could be further improved if additional predictors are added. For example, it is hypothesized that a customer's education might provide additional insight, as those who are more educated might read more. Additionally, age of the customers would also be helpful, as the younger generation might have different reading and buying patterns than the older generation. Finally, most recent login to the online bookstore might also shed additional light for the model on who would actually be more likely to purchase.

References

- Your textbook :)