Northwestern University

IEMS 462-1 Course Project

An Analysis of Sales Data

Author:
Joel Eliason
Ethan Rucinski
Huazhen Zhao

Professor:
Ajit Tamhane

March 16, 2018

Executive Summary

This project aims at predicting the sales amount using the sales data from Sep 1, 2012 to Dec 1, 2012. We are interested in investigating two specific questions: who will purchase and how much they will purchase.

Through various predictive models, the most significant variables for the probability of the responding purchase amount are life-to-date on fall orders, number of days since the last purchase and interactions between the order data between this year and last year's orders.

As for predicting specific purchase amount, the sales amount from the last two years serves as a good positive indications of customer's responding purchases amount between Sep 1, 2012 and Dec 1, 2012. In addition, people with more purchase amount since their first purchase tend to purchase more during the three months.

As a result of our models, we select top 1000 expected customer with their actual purchase to be $51615.37. Based on their past purchase records, we recommend the management to focus on maintaining good relationships with people who make relatively large purchases amount in the past, whose contribution would make a majority of sales amount.

Contents

1 Introduction

Given the sales data from Sep 1, 2012 to Dec 1, 2012, we explored two questions: based on past record, which customers are more likely to purchase during the three months. Also, among those who did purchased from Sep 1, 2012 to Dec 1, 2012, which variables of their record could help us to explain the differences in their purchase amount.

To investigate those two questions, we organized the report into five parts: Section 2 – Exploratory Analysis and Data Preprocessing, Section 3 – Model Fitting, Section 4 – Model Validation, Section 5 – Conclusions, and the Appendix.

In Section 2, the Exploratory Analysis and Data Preprocessing section, we examine the nature of data by performing various diagnostic test on the variables. We also correct inaccurate records from the dataset. Particularly, convert dates to numeric or categorical variables for model fitting, create relevant interaction variables, assess the distributions of the variables and check for mismatches between variables that were measuring the same thing (e.g. mismatches between sales and orders data).

Section 3, the core of our report, details how we build up the prediction models. It is presented with two subsections: Logistic Regression and Multiple Regression. We utilized both stepwise regression methods and simultaneous selection/regression methods in order to find the most parsimonious models without losing accuracy. In Logistic Regression, we would explain the overall procedure and the details of how we conduct predictors selection, interactions, model fitting, correct classification rate, model diagnosis. In Multiple Linear Regression Model, in addition to similar procedures of logistic regression, we also do data transformation, such as log transformation.

Section 4 tests our models built in Section 3 on the validation set, including many key performance measures. Section 5 concludes the whole investigation by noting important

discoveries, comments, and directions for further improvement.

2 Exploratory Analysis and Data Preprocessing

As a group, we decided to divide up all the potential predictor variables amongst the group members for exploratory analysis and preprocessing. Each of us then performed various diagnostic tests on the variables we were assigned to look for missing values and outliers, convert dates to numeric or categorical variables for model fitting, create relevant interaction variables, assess the distributions of the variables and check for mismatches between variables that were measuring the same thing (e.g. mismatches between sales and orders data).

A barplot of targdol (see Figure 1a)) did not reveal anything strange in the distribution of this variable. Furthermore, there were no missing values in the data (as assessed by is.na() in R).

Among the first five predictor variables, our first step was to reformat datelp6 and datead6 into Date variables in R, and create a new variable, dm_lp, that simply contains the day and month of each datelp6 variable. A barplot of dm_lp can be seen in Figure 1b. As can be seen, this distribution is hardly uniform across all days in the calendar year, with most observations falling principally into March 1 or November 15. As we had no way of 'redistributing' these binned variables, we simply decided to bin all of the variables into 6 month bins, creating a new variable called lp6_bin, which replaced the datelp6 variable in our analysis. Next, many of the observations in variable lpuryear did not match the year in variable datelp6 - to remedy this, we assumed that the date reported in datelp6 was correct and created a new variable (lpuryear2) that simply reported the year in the datelp6 variable. Next, a barplot of datead6 (Figure 1c) again reveals a quite

nonsymmetric distribution of dates. Furthermore, we noticed that many of the dates in datead6 were *after* the corresponding datelp6 - an impossibility. However, as datead6 seemed a priori a not very significant variable in predicting future sales, we decided to make no changes to this variable unless it did show up as a significant predictor. Next, in checking the distributions of slstyr and slslyr (Figures 1d and 1e), we noticed nothing odd about the distribution, so made no corrections here. However, we noticed that some observations that reported no sales simultaneously reported nonzero orders. We remedied this by assuming that the sales data were correct and simply setting the orders to zero for those observations. Lastly, we created a new recency variable, which was simply the number of days since the last purchase (essentially creating a coding for the logistic regression). We made this available both as a numeric variable and as a categorical variable (the latter created from the lp6_bin variable, rather than datelp6). Furthermore, we created some consistency variables that were interactions between the order data: consistent1, consistent2 and consistent3 were interactions between the order data between this year and last year's orders; consistent1 and the year before's orders; and consistent2 and 3 years ago, respectively. Similarly, ord1b, ord2b and ord3b coded 0 or 1 if there were orders in those years, and cons1b, cons2b and cons3b were interactions between ord1b, ord2b and ord3b, constructed similarly to consistent1,2 and 3.

Next, we sought to fix the last purchase date data. As stated above, we noticed that we could bin our purchase dates in 6 month groups. However, not all of the data for the last purchase date aligned with the data we had on how many purchases were made in a particular year. In some cases, we had last purchase dates which did not reflect purchases made in more recent years. We sought to correct the last purchase year variable, as a result. However, we also noted that we wouldn't necessarily know the last purchase date or month, if we needed to update the last purchase year. From the complete data points

which we had collected in 6 month bins, we found the distribution of purchases in each bin for each year. Then, when we updated the last purchase years for our observations, we chose the last purchase bin to be randomly selected based on the previous observed probability of being in each bin from that year. The resulting data set was what we used for the rest of our analysis in this project.

3 Model Fitting

In this section, we detail our model fitting approaches. We utilized both stepwise regression methods and simultaneous selection/regression methods in order to find the most parsimonious models without losing accuracy. In the section below, we analyze the steps that we took to fit these models and find the most significant models, as well as the relevant model diagnostics.

3.1 Predict Responding Probabilities: Logistic Regression

After removing all extraneous variables from the data set, restricting the data set to the train observations and making sure all necessary interactions and new variables were present (e.g. the recency and consistency variables), I decided to first fit the data set using LASSO regression with cross validation, using the glmnet package in R. This was a very straightforward and fast regression to perform, as it does a simultaneous fitting/selection. Furthermore, it allows diagnostics with the deviance, correct classification rate and ROC curve. I also attempted to use the bestglm package - this, however, had a very long run time (compared to the minute or two with glmnet), and so I here report model selection results only from the LASSO regression.

When doing LASSO logistic regression, we have the option of using a few different

Table 1: Comparison of different penalties in LASSO regression

Method	CCR	$p*$	#Nonzero	AUC
MAE	0.9357346	0.48	22	0.7530062
Misclass	0.9353901	0.31	7	0.7490637
AUC	0.9357346	0.46	22	0.7545231
Deviance	0.9356269	0.48	15	0.7484009
Full model	0.9357992	0.06	27	0.7469825

methods to find the best fitting and most parsimonious model: minimizing the deviance, minimizing the misclassification rate, minimizing the mean absolute error or maximizing the AUC (area under the ROC curve). In my model fitting, I performed the LASSO logistic regression with all 4 loss functions, and then used the correct classification rate (along with the confusion matrix), parsimony of the model and AUC as model diagnostics in how I decided to finally select a model. Furthermore, since there is no consensus in the research community on best practices for calculating standard errors in penalized regression (as in LASSO), I leave out this diagnostic (and significance testing by z-value of coefficients) at this particular stage of the model fitting. In Table 1, for each of these methods, I report the correct classification rate, the number of nonzero coefficients, the $p*$ value used for the CCR and the AUC for the model.

The correct classification rate is calculated by finding the value of $p*$ for which CCR is maximized, then computing the confusion matrix for this particular classification and calculating the CCR as normal. See Figures 2b-2d for the $p*$ vs CCR plots. Furthermore, the ROC curves for each of these methods can be seen in Figures 3a-3d.

All of these models have very similar CCR and AUC. However, they differ quite drastically in the number of nonzero variables that they all have. As these estimates were obtained by penalized regression methods, the coefficients have no exact or asymptotic distribution, and thus we cannot confidently calculate the standard error and perform

significance tests. Because of this, as my next step in model diagnostics, I decided to fit a standard logistic regression (GLM) using only the variables selected in each LASSO method and the significance of the variables for the most parsimonious model (the model produced by the misclassification penalty, with 7 variables). The full summary of these fits can be found in the Appendix; however, I will note some of the results here. Once the model had been fit using the 7 variables, we noted that there still remained some insignificant variables. Paring the model of those variables brought us to a model with 3 variables: falord, recency_numeric and cons1b. These variables are all highly significant (I briefly experimented with adding each the removed variables in turn back to this sparse model; however, the additions simply produced a model with less significant variables). Furthermore, the CCR of this model is 0.9357346, very comparable to the maximum CCR in Table 1 (this is with using $p*$=0.43). It should be noted that its AUC (0.7469825) is very slightly lower than any of the above AUCs.

An overall significance test of the model yielded $D_0^2 - D^2 = 2460$ on 3 d.f. - this is clearly a significantly better model than the null model (again, deviances can be found in the Appendix), allowing us to reject the null hypothesis that all coefficients are zero. A comparison of this 3-variable model with the 'full' model that has 7 variables yields a difference of deviances of 11 on 4 d.f., which is just barely significant at the 95% level. Lastly, a comparison of the 3-variable model with the actually full model (with all of the variables) yields a chi-squared statistic of 367 on 21 variables, which is certainly very significant. However, because its classification rate is quite comparable to these fuller models (if not better), and because of its high levels of parsimony and significance of variables in comparison with the other models, this is the model that we ended up using.

3.2 Predict Purchase Amount: Multiple Regression

To perform our multiple regression model creation, I used stepwise regression. First, I selected from the data observations where `TARGDOL` was non-zero. Then, I added the same interaction variables as above and removed obsolete predictors, restricted the data to the training set, and created a full model with all the remaining predictors.

Next, I ran the stepwise regression and found a resulting model with an adjusted R^2 value of 0.0844. The output for this model can be found in the in the appendix.

Next, I performed basic diagnostics on this model. In plotting the residuals vs. fitted values, and the normal Q-Q plot, both found in the appendix, there are three repeat-offender outliers (Row ID 37839, 38283, and 90895) that I removed from the data before going forward. After removing these data points, I ran stepwise regression again and this time was able to achieve an adjusted R^2 value of 0.1129. However, in the normality plot, included in the appendix, it is clear that the residuals violate the normality assumption.

To proceed, I applied a log transformation to `TARGDOL` and ran a stepwise regression again. In the resulting model, the F-Statistic decreases slightly from the last stepwise model, but the normality assumption holds based on the resulting Q-Q plot. This model had an adjusted R^2 statistic of 0.1098.

`Model 1(StepReg3): log(targdol) ~ slstyr + slslyr + sls2ago + slshist + ordhist + consistent2 + ordtb + ordlb + ord2b + ord3b + cons1b`

After performing the stepwise regressions, I tried to use LASSO regression, but the resulting model did not prove to be more powerful than the last stepwise regression.

`(LASSO): targdol ~ slstyr + slslyr + sls2ago + slshist + ordhist + falord + consistent3 + ordlb + ord3b + cons2b + cons3b`

After examining plots of `LASSO`, I noticed that the Q-Q plot is heavy-tailed in Model 2 which indicates a log transform on response variable. I preformed log transform on

TARGDOL and the normality assumption is satisfied based on the resulting Q-Q plot.

(log-LASSO): log(targdol) $\sim$ slstyr + slslyr + sls2ago + slshist + ordhist + falord + consistent3 + ordlb + ord3b + cons2b + cons3b

However, not all predictors are significant in Model LASSO and log-LASSO. In order to find the parsimony model for the two Models, I delete all variables that are not significant.

Model 2(parsimony): targdol $\sim$ slstyr + slslyr + sls2ago + slshist + ordhist

Then I performed log transform to satisfy normality assumption. The plots can be found in appendix.

Model 3(log-parsimony): log(targdol) $\sim$ slstyr + slslyr + sls2ago + slshist + ordhist

After performing a series of stepwise regressions and transforms we are able to select Model 1,2 and 3 to be the candidate models for validation.

4 Model Validation

4.1 Validation for Logistic Regression

We conduct validation for our logistic regression model. We apply the logistic model to our testing dataset with the cutting off probability 0.43, we could get our confusion matrix as follows

Confusion	FALSE	TRUE
0	43310	2908
1	85	145

The correct classification rate rate is 93.57%

4.2 Purchase Amount Validation

To validate the goodness of our multiple regression results, we separate test set observations with `TARGDOL` > 0, apply models and calculate validation Mean Square Error of Prediction (MSEP) on testing data. Table 2 presents these results:

Model	Test MSEP	Test RMSE
1 (stepReg3)	3872.069	62.22595
2 (parsimony)	2541.693	50.41521
3 (log-parsimony)	4164.727	64.53469

Table 2: Validation Results for Multiple Regressions

As we can see from the Table 2, the mean square error of two models are ranging from 4165 to 2542. The data is scatter widely and 2542 is not a bad fit. Therefore, we are going to chose Model 2 to predict the purchase amount on the entire validation set since it performs the best.

4.3 Combined Validation

In this section, we combine both parts of our models, binary logistic classification and multiple regression results, to calculate expected purchase amount for each sample of data in the validation set. Table 2 contains MSEP result for the entire dataset. Also presented at the bottom rows are the actual sum of `TARGDOL` values for 1,000 customers with highest `E(TARGDOL)`; as a reference, we also include the theoretical maximum amount of this measure by summing the highest 1,000 `TARGDOL` values across the whole validation samples. The result is decent because the RMSE values do not differ too much from corresponding values in Table 2, indicating good extensibility and consistency; also the payoff calculated by summing actual `TARGDOL` values for one thousand customer with highest expected purchase value is over the half of the theoretical maximum.

Model	MSEP	RMSE
3 (parsimony)	2900.265	53.85411
Actual Purchase by the 1k highest `E(TARGDOL)`:		$51615.37
Sum of highest 1k `TARGDOL`'s:		$92072.24

Table 3: Combined Validation Results

5 Conclusion

The strategy we adopted is a two-step model fitting approach: Logistic Regression and Multiple Regression. They are used to predict the probability for each customer and predict the amount of purchases.

Logistic Regression shows that the significant predictors are `falord, recency-numeric, ordtb*ordlb` which `recency-numeric` is the number of days since last purchase and `ordtb, ordlb` are the binary variables for `ordtyr2` and `ordlyr2`. More specifically, people who have more orders in the past two years have a higher responding probability. Not surprisingly, the number of days since last purchase is negatively correlated with responding probability. This is reasonable because recent buyers are more likely to respond.

Multiple Linear Regression shows that `slstyr, slslyr, sls2ago, slshist, ordhist` are significant predictors. The past sales amount is also significant in predicting purchase amount from designated customers. Customers who have a higher purchase amount before will contribute more in the future. However, The value of coefficients of Multiple Regression model gives us another perspective. Expect for the positive correlation with `slstyr, slslyr, sls2ago, slshist` , it's worth noting that `ordhist` negatively correlates with sales amount. This is because the number of purchases are less for the expensive merchandise.

After running models and data analysis, we find that our relevant data are still limited. For example, we lack the knowledge of the average purchase interval of each customers,

we also lack the unit price of each merchandise.

Appendix

GLM Fits for Logistic Regression

Full Model

```
Call:
glm(formula = gzro ~ ., family = binomial, data = train)


Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0060  -0.3911  -0.2926  -0.2071   3.7221


Coefficients: (2 not defined because of singularities)
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      1.067e+03  1.690e+02   6.313 2.74e-10 ***
slstyr           1.134e-04  6.601e-04   0.172  0.86356
slslyr          -3.014e-04  6.672e-04  -0.452  0.65144
sls2ago         -4.937e-04  7.846e-04  -0.629  0.52923
sls3ago          3.214e-04  6.365e-04   0.505  0.61355
slshist         -2.088e-04  2.854e-04  -0.732  0.46442
ord2ago          1.182e-01  7.257e-02   1.629  0.10334
ord3ago          4.459e-02  7.687e-02   0.580  0.56185
ordhist          5.401e-02  2.376e-02   2.273  0.02301 *
falord           1.773e-01  2.449e-02   7.241 4.45e-13 ***
sprord                  NA         NA      NA       NA
```

```
train                    NA         NA      NA       NA
lpuryear2       -5.347e-01  8.520e-02  -6.276 3.48e-10 ***
lp6_bin          1.325e-01  4.163e-02   3.182  0.00146 **
ordtyr2          1.189e-01  8.114e-02   1.465  0.14280
ordlyr2          1.556e-01  7.458e-02   2.087  0.03690 *
recency_numeric -9.291e-03  5.242e-04 -17.727  < 2e-16 ***
recency_factor   1.413e+00  9.027e-02  15.652  < 2e-16 ***
consistent1     -1.819e-01  6.960e-02  -2.613  0.00897 **
consistent2      2.949e-03  5.177e-02   0.057  0.95458
consistent3     -5.720e-02  2.951e-02  -1.938  0.05263 .
ordtb            1.285e-01  1.340e-01   0.959  0.33752
ordlb            6.513e-02  1.159e-01   0.562  0.57419
ord2b            4.786e-02  9.671e-02   0.495  0.62071
ord3b            1.555e-01  1.010e-01   1.540  0.12356
cons1b           6.031e-01  1.486e-01   4.059 4.92e-05 ***
cons2b          -2.840e-02  1.813e-01  -0.157  0.87552
cons3b           2.902e-01  1.893e-01   1.533  0.12529
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22660  on 45799  degrees of freedom
Residual deviance: 19833  on 45774  degrees of freedom
AIC: 19885
```

```
Number of Fisher Scoring iterations: 6
```

7 Variable Model

```
Call:
glm(formula = gzro ~ ordhist + falord + recency_numeric + ordtb +
    cons1b + cons2b + cons3b, family = binomial, data = train)


Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1717  -0.4031  -0.3151  -0.2167   3.2482


Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -2.2299390  0.0648463 -34.388  < 2e-16 ***
ordhist          0.0338992  0.0174171   1.946   0.0516 .
falord           0.2311613  0.0221234  10.449  < 2e-16 ***
recency_numeric -0.0008628  0.0000448 -19.259  < 2e-16 ***
ordtb            0.0322125  0.0587134   0.549   0.5833
cons1b           0.4966104  0.0805749   6.163 7.12e-10 ***
cons2b           0.0634109  0.1357059   0.467   0.6403
cons3b           0.2542545  0.1576817   1.612   0.1069
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 22660  on 45799  degrees of freedom
Residual deviance: 20189  on 45792  degrees of freedom
AIC: 20205

Number of Fisher Scoring iterations: 6
```

Parsimonious Model

```
Call:
glm(formula = gzro ~ falord + recency_numeric + cons1b, family = binomial,
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1549  -0.3970  -0.3175  -0.2124   3.2729

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -2.194e+00  4.257e-02 -51.531   <2e-16 ***
falord           2.771e-01  1.052e-02  26.349   <2e-16 ***
recency_numeric -8.862e-04  3.473e-05 -25.519   <2e-16 ***
cons1b           6.005e-01  6.197e-02   9.689   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22660  on 45799  degrees of freedom
Residual deviance: 20200  on 45796  degrees of freedom
AIC: 20208

Number of Fisher Scoring iterations: 6
```

Model 1: StepReg3

```
Call:
lm(formula = log(targdol) ~ slstyr + slslyr + sls2ago + slshist +
    ordhist + consistent2 + ordtb + ordlb + ord2b + ord3b + cons1b,
    data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-2.84037 -0.52313 -0.02138  0.48924  2.89100

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.5243734  0.0287922 122.407  < 2e-16 ***
slstyr       0.0033740  0.0004633   7.283 4.12e-13 ***
slslyr       0.0016267  0.0004323   3.763 0.000171 ***
sls2ago      0.0011085  0.0004590   2.415 0.015787 *
slshist      0.0010520  0.0001315   8.002 1.72e-15 ***
```

```
ordhist     -0.0396164  0.0083100  -4.767 1.95e-06 ***
consistent2 -0.0355857  0.0168629  -2.110 0.034913 *
ordtb       -0.1999166  0.0433381  -4.613 4.13e-06 ***
ordlb       -0.1456296  0.0433984  -3.356 0.000801 ***
ord2b       -0.0748146  0.0385639  -1.940 0.052469 .
ord3b       -0.0482541  0.0339862  -1.420 0.155763
cons1b       0.1656581  0.0589199   2.812 0.004961 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.746 on 3084 degrees of freedom
Multiple R-squared:  0.1129,Adjusted R-squared:  0.1098
F-statistic:  35.7 on 11 and 3084 DF,  p-value: < 2.2e-16
```

Model 2: Parsimonious Model

```
Call:
lm(formula = targdol ~ slstyr + slslyr + sls2ago + slshist +
    ordhist, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-163.65  -24.19  -11.58    9.35  487.97

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 42.282054   1.219008  34.686  < 2e-16 ***
slstyr       0.118285   0.020508   5.768 8.82e-09 ***
slslyr       0.047163   0.019574   2.409   0.0160 *
sls2ago      0.036613   0.020853   1.756   0.0792 .
slshist      0.078541   0.006858  11.453  < 2e-16 ***
ordhist     -3.510698   0.361927  -9.700  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 42.83 on 3090 degrees of freedom
Multiple R-squared:  0.1092,Adjusted R-squared:  0.1078
F-statistic: 75.79 on 5 and 3090 DF,  p-value: < 2.2e-16
```

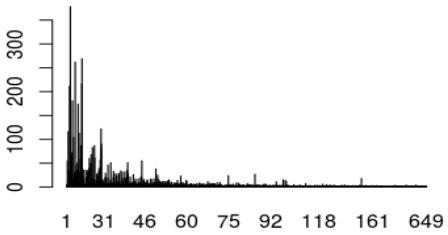
Model 3: log-parsimony

```
Call:
lm(formula = log(targdol) ~ slstyr + slslyr + sls2ago + slshist +
    ordhist, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-2.61647 -0.52152 -0.02265  0.48342  2.86439

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.4446710  0.0213229 161.548  < 2e-16 ***
```

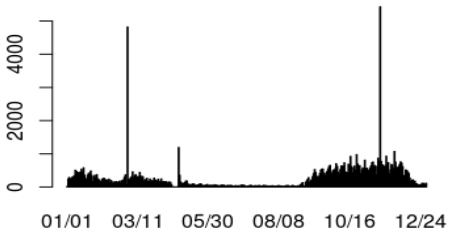
```
slstyr       0.0021733  0.0003587   6.058 1.54e-09 ***
slslyr       0.0009626  0.0003424   2.811  0.00496 **
sls2ago      0.0004606  0.0003648   1.263  0.20675
slshist      0.0013102  0.0001200  10.922  < 2e-16 ***
ordhist     -0.0621122  0.0063308  -9.811  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7491 on 3090 degrees of freedom
Multiple R-squared:  0.1037,Adjusted R-squared:  0.1023
F-statistic:  71.5 on 5 and 3090 DF,  p-value: < 2.2e-16
```

AUC Plots

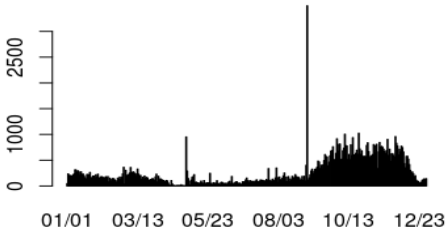
Model diagnose plots for Multiple Regression



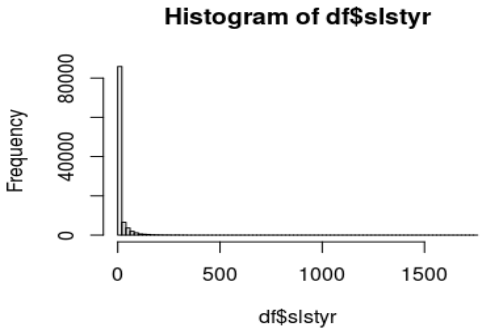
(a) Histogram of targdol



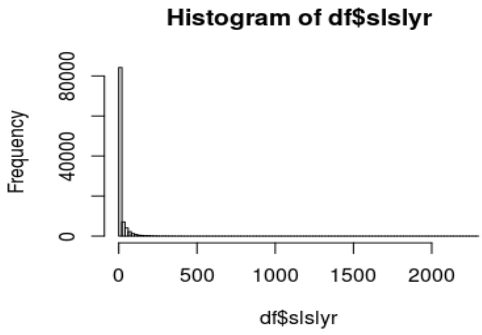
(b) Histogram of dm_lp



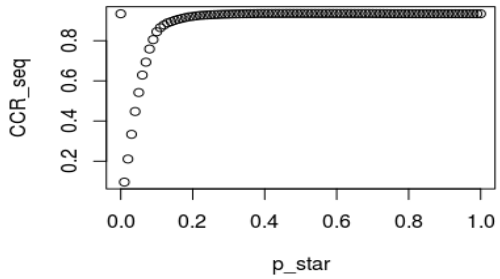
(c) Histogram of datead6



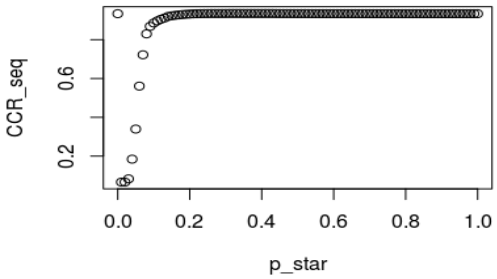
(d) Histogram of slstyr



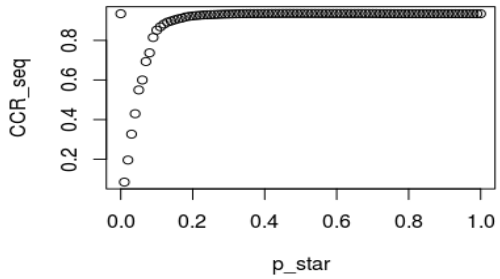
(e) Histogram of slslyr



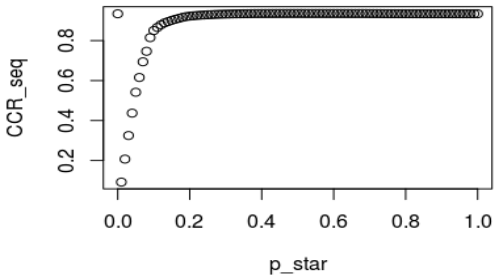
(a) CCR plot for AUC method



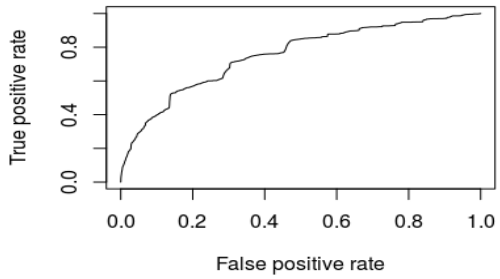
(b) CCR plot for misclassification method



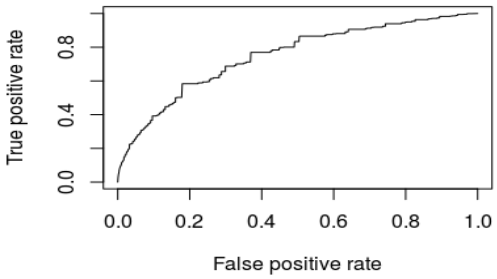
(c) CCR plot for the deviance method



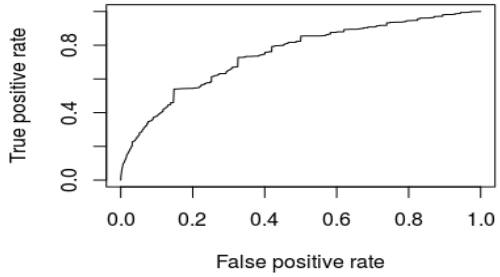
(d) CCR plot for the MAE method



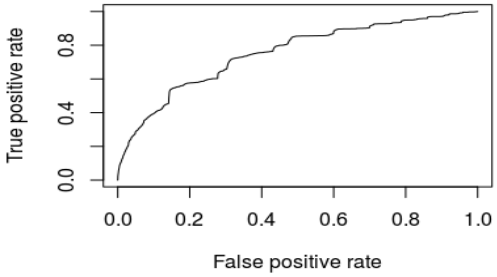
(a) ROC curve for AUC method



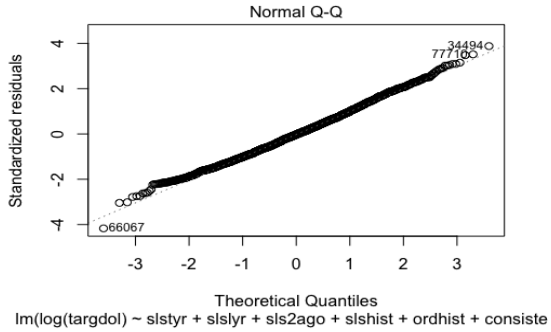
(b) ROC curve for the misclassification method



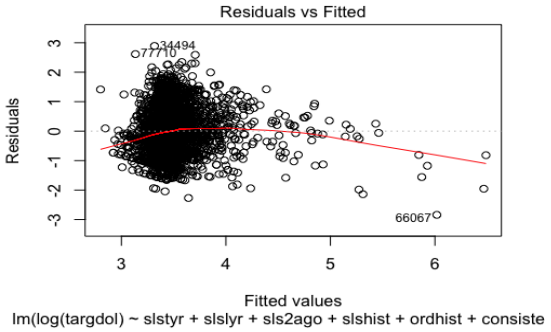
(c) ROC curve for the deviance method



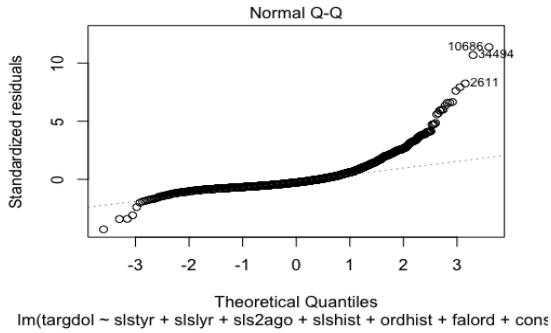
(d) ROC curve for the MAE method



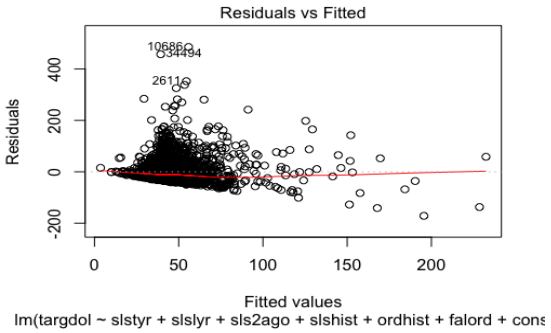
(a) Normal Q-Q plot for Model 1



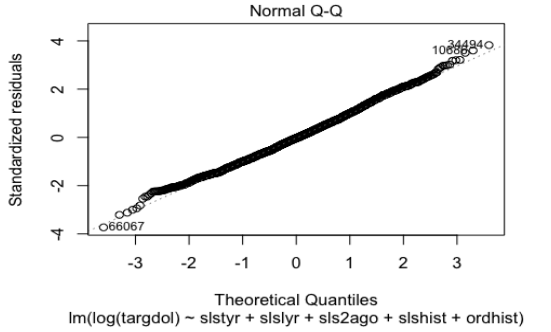
(b) Residual & Fitted plot for Model 1



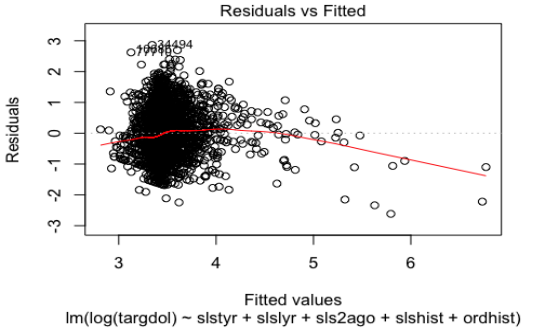
(c) Normal Q-Q plot for Model 2



(d) Residual & Fitted plot for Model 2



(e) Normal Q-Q plot for Model 3



(f) Residual & Fitted plot for Model 3