Documentation of the Payment Delays Project

Ivana Stanova, Lenka Stastna

10 1 2021

Contents

1 Business Understanding

The main topic of this data science project is to optimize the collection process. Sponsor of the project is Mr. Jiří Procházka, who decided on the scope of the project, provided data and helped define project deliverables.

First of all, it was important to understand the collection process. Clients are paying a certain amount due to a certain date. If the client delays his payment, certain collection procedure takes place because of the costs that rise for the company. There are three actions taking place under different conditions. If the payment is delayed for 21 - 69 days, first action is taken. Second action is taken if the delay is between 70 - 139 days and the third action is taken for delay greater than 140 days.
Our goal was to create three models. First model predicts whether the customer will be delayed in payment for 21+ days. Second model predicts whether the customer will be delayed in payment for 140+ days. Lastly, third model should estimate the average number of days delayed if the client would exceed the first action. All three models were defined baseline accuracy, to which created models were compared.

As one of the deliverables, project documentation describes project flow. Project consisted of three major steps: Data preparation, Modeling, Evaluation. All of the phases are described in detail with visualized results and findings in this document.

We chose to use R studio to prepare the data, create models and evaluate, RMarkdown was used to create documentation of the project.

2 Data Preparation

2.1 Data Understanding

2.1.1 Data Description Report

The initial data was provided in a comma-separated values file, and was loaded and processed using the R programming language. Dataset used in this project contains 2 353 012 observations and 24 variables. Out of the 24 variables, 13 are of factor datatype, 9 are numeric and 2 are dates. All columns from the initial dataset were converted to the correct datype according to the data description file, which was provided. Column *payed_ammount* was replaced by column *paid_amount*. Column *payment_date* originally contained some blank fields, which were subsequently filled in as NA. We also created a new feature *delay* at the beginning of our work as the target variable. Variable stands for the difference between *payment_date* and *due_date*.

Column name	Description	Type	Values
contract_id	Unique identificator of the contract	Int	{1,2,3,...,N}
payment_order	Order of the payment	Int	{1,2,3,...}
due_date	Payment deadline	Date	YY/MM/DD
payment_date	Date of the payment	Date	YY/MM/DD
product_type	Type of the product	Factor	{1,2,3,4,5}
contract_status	Contract status	Factor	{1,2,3,4,5,6,7,8,9}
business_discount	Business discount provided	Factor	{0,1}
gender	Gender	Factor	{1,2}
marital_status	Marital status	Factor	{1,2,3,4,5,6}
number_of_children	Number of children	Int	{1,2,3,...}
number_other_product	Number of other products	Int	{1,2,3,...}
clients_phone	T/F if the client filled in home phone	Factor	{True, False}
client_mobile	T/F if the client filled in mobile phone	Factor	{True, False}
client_email	T/F if the client filled in email address	Factor	{True, False}
total_earnings	Earning bucket	Factor	{level1,..., not_declared}
birth_year	Birth year of the client	Int	{1990,1991,...}
birth_month	Birth month of the client	Int	{1,2,3,...}
living_area	Region of the client home address	Factor	{1,2,3,...}
different_contact_area	T/F if the client filled different home and contact address	Factor	{True, False}
kc_flag	T/F if the client does not home local citizenship	Factor	{True, False}
cf_val	If the special measure during the underwriting was applied	Numeric	{-N,...,N}
kzmz_flag	T/F if the client filled in employer	Factor	{True, False}
due_amount	Installment what should be payed	Numeric	(0,...)

Column name	Description	Type	Values
payed_amount	What was payed at a certain date	Numeric	(0,. . .)
delay	Difference between payment_date and due_date	Int	{-N,. . . ,N}

2.1.2 Attribute correlations

We computed correlation coefficients between all possible pairs of numeric variables, see Figure 1, and discovered strong positive correlation between *due amount* and *paid amount*. This could be due to the fact that in the event that the installment has already been paid, the due amount and the paid amount would assume the same value. Correlation between the remaining pairs of numeric variables was either nonexistent or negligible.
Then, the significance of correlation between due amount and paid amount was tested using Pearson's product moment correlation coefficient. The pair of attributes was found to be significantly correlated with a correlation coefficient of 0.76 and p-value less than 2.2e-16.
Relationship between categorical variables was tested using chi-squared test with the significance level of 0.05. All significantly correlated pairs of variables can be accessed in "categorical_rel" dataframe.

2.1.3 Basic statistics

Basic statistics computed for numeric variables can be located in Table 6. Frequency, relative frequency and relative cumulative frequency were computed for each categorical variable and its categories. All frequency tables can be located in the "frequencies" list.

2.2 Data Exploration Report

Distribution of numeric and categorical variables was visualized using boxplots, density plots and histograms, see Figure 2 and Figure 3. Next, we performed bivariate analysis of continuous variables with respect to categorical variables on selected pairs of features. The results for depedence on gender can be accessed in tables "data_GPA" (dependence of paid amount on gender), "data_GD" (Statistical dependence of delay on gender), "data_GDA" (Statistical dependence of due amount on gender). We found out that business discount applies only to product type 1, as seen in Figure 4. Product type 1 displayed the highest median delay at around 25 days, followed by products 2, 3, 4, all with median delay at around 10 days. Product type 5 displayed the lowest median delay. The results did not significantly differ between the genders.

2.3 Data Quality Report

2.3.1 Data coverage

Next step consisted of the data coverage and plausibility analysis.
We did not find the results surprising, but as an example, we have chosen a couple of plots, that indicate interesting data distribution. We, for example, found out that clients mostly order the product type 1, contracts are mostly in status 5 or that most of the payments have a discount. We also discovered, that the marital status of the clients is mostly number 3 and they have most frequently no children. Clients also very frequently do not provide information about their earnings and they usually ordered 1 other product. Although the distribution of values across factor variables is not even, we do not think, the findings have to be analyzed closely.All the mentioned findings can be seen on visualizations in Figure 5.

2.3.2 Missing values

Exploring the NA values in the dataset, we found out, that 4 attributes had almost the same percentage of missing values, as can be seen in the statistics 4.
Attributes *kc_flag*, *living_area*, *cf_val* and *different_contact_area* have the most missing values, almost 20 %, whereas *payment_order* has around 3,5 % and *payment_date* and *delay* have the same percentage, almost 0,5 %.
Using a different visualization, that can be seen in Figure 6 or Figure 7, we discovered, that the four attributes with the highest percentage are not missing at random but almost all at the same time.
We found out, that contract_id together with payment_order were not creating a unique key of the payment. One payment was divided into multiple parts, which was also causing problem with NA values in the four attributes. Data in the four attributes were not copied into other parts of a payment, but were present in just the first payment part.
We decided to unify the payment parts into only one payment by summarizing the paid amount of all the parts and using the payment_date of the last paid part. Thanks to the unification, the amount of NA values has markedly decreased.
Secondly, we dealt with the NA values in payment_order and payment_date. Since it was only less then 4 % of the dataset, and it was not possible to substitute the values, we decided to delete the rows.

2.4 Feature engineering

We decided to add new features to create higher-accuracy models. As already have been mentioned, we firstly computed a numeric feature *delay* counting the difference of days between *payment_date* and *due_date*. We selected this variable as our target variable.

Since we are creating two classification models deciding whether a new payment will be delayed for more than 21 days or more than 140 days, we created 2 new factor features *delay_21_y* and *delay_140_y*. Value is set to 1 if the payment delay is greater than 21 or greater than 140 days.

We also created a new numerical feature *delay_indiv* counting the mean delay for the whole client´s history. We also computed 2 new numerical features, *delay_indiv_21* and *delay_indiv_140* counting number of delayed payments (21, 140 days) for the whole client's history.

Lastly, numerical features *mean_delay_1m* ,*mean_delay_3m*, *mean_delay_6m*, *mean_delay_12m* are computing the mean delay for the last 1/3/6/12 months in the client´s history.

2.5 Exploratory analysis of the new features

Adding new variables, we started to work with 34 variables. We created 2 factor variables and 8 numerical variables

2.5.1 Data description report

Column name	Description	Type	Values
delay_21_y	T/F if the delay is more than 21 days	Factor	{True, False}
delay_140_y	T/F if the delay is more than 140 days	Factor	{True, False}
delay_indiv	Mean delay for the whole client's history	Int	{-N,. . . ,N}

Column name	Description	Type	Values
delay_indiv_21	Cumulative sum of payments delayed for more than 21 days by contract	Int	{1,2,3,. . . ,N}
delay_indiv_140	Cumulative sum of the payments delayed for more than 140 days by contract	Int	{1,2,3,. . . ,N}
mean_delay_1m	Average payment delay for the last month	Int	{-N,. . . ,N}
mean_delay_3m	Average payment delay for the last 3 months	Int	{-N,. . . ,N}
mean_delay_6m	Average payment delay for the last 6 months	Int	{-N,. . . ,N}
mean_delay_12m	Average payment delay for the last 12 months	Int	{-N,. . . ,N}

2.5.2 Basic statistics

Basic statistics computed for new numeric variables can be located in Table 7. First of all we focused on attribute *delay*, as our target attribute. On Figure 8, we provided 4 different plots visualizing delay variable. As we can see, the values fluctuate mostly around 0. Although values range from -1673 to 2787, delay median is 17 and delay mean is 22.13.

We also computed some basic statistics for the other added attributes. Results can be seen on Figure 9, where we visualized the distribution of delay_indiv and mean delay variables.

On Figure 11 we can see the distribution of variable *delay_indiv_21* and *delay_indiv_140*. Delay greater than 140 is present only in a few payments, whereas delay greater than 21 days is more common.

Lastly, we also analyzed factor variables, *delay_21_y* and *delay_140_y*. Results are visualized on Figure 10, where almost half (48,2 %) of all the payments were delayed for more than 21 days and only 9 % of all the payments have delay larger than 140 days.

2.5.3 Missing values

As can be seen on Figure 5, newly-created features also contain NA values. The highest percentage of missing values has attribute *mean_delay_12m*, almost 55 %. Together with *mean_delay_6m*, *mean_delay_3m* and *mean_delay_1m*, they are the only new attributes holding NA attributes.
It is not surprising, that these attributes have the highest percentage of NAs, since they compute results only every 12/6/3/1 months. We decided to replace the NA values by 0, so they can be later used in the modeling part. We assume, that this step should not influence the models.

3 Modeling

3.1 Prediction model (21+ days)

The goal of this classification task was to predict if the customer will be delayed in payment for 21 and more days. The expectation of our sponsor for the model was for area under the curve (AUC) to exceed 0.7.
We decided to use penalized logistic regression model (elastic net). The simple model was chosen due to the team having little prior experience with data science. First, we excluded *delay_140_y* (as it was deemed irrelevant for this part of modeling), delay and payment_date (because they caused 100% accuracy). All

missing values in *mean_delay_1/3/6/12m* features were replaced by 0.
Due to *living_area* having too many levels, we used weight of evidence (WOE) to split living_area into a set of bins (by combining categories with similar WOE) based on similarity of *delay_21_y* variable distribution. For this action, we transformed *delay_21_y* into a numeric datatype. The optimal binning for *living_area* was found to be 4 and the original variable was replaced by the binned version, see Figure 12. Finally, *delay_21_y* was transformed back to factor.

We calculated information value for the independent variables, with dependent variable being *delay_21_y*. We discovered that *mean_delay_1m*, *mean_delay_3m*, *contract_id*, *mean_delay_6m* and *delay_indiv* provided the most information value.

The dataset was then split into training (60%), validation (20%) and test (20%) datasets. We used stratified sampling to avoid missing classes in training data. Hypergrid was defined to tune the parameters. After fitting the model, variable importance was calculated and we discovered that *mean_delay_1m*, *delay_indiv_21*, *contract_status5*, *delay_indiv_140* and *delay_indiv* provided the best results. Afterwards, we used hold-out to determine cutoff and to find the best values for alpha and lambda using training data. The optimal cutoff value which maximized both specificity and sensitivity was found to be 43.7%. Using alpha found in the previous step, we used 5-fold cross-validation to find optimal value for lambda (using validation data).
The accuracy of created model is 88.66% and the AUC is 0.946, see Figure 13. For confusion matrix, sensitivity and specificity, please refer to the table below.

```
        Confusion matrix
preds2       0      1
    0 135752  18293
    1  14608 121512

Sensitivity : 0.8692
Specificity : 0.9028
```

Maximum accuracy for both cross-validation and hold-out method was achieved with hyper parameters *alpha = 0* and *lambda = 0*. Analyzing the hypergrid, we discovered that maximum accuracy was connected to lambda always being 0 and alpha then becoming irrelevant and ranging from 0 to 1. The best results were achieved with no penalization.

We calculated the top decile lift to be 2.064 by ordering the data by the predictors and computing the proportion of positives in the top 10%. For lift curve please refer to 16.

3.2 Prediction model (140+ days)

The goal of this classification task was to predict if the customer will be delayed in payment for 140 and more days. The expectation was for AUC to exceed 0.7.

We decided to use penalized logistic regression model (elastic net). The simple model was chosen due to the team having little prior experience with data science. First, we excluded *delay_21_y* (as it was deemed irrelevant for this part of modeling), delay and payment_date (because they caused 100% accuracy). All missing values in *mean_delay_1/3/6/12m* features were replaced by 0.
Due to *living_area* having too many levels, we used weight of evidence (WOE) to split *living_area* into a set of bins (by combining categories with similar WOE) based on similarity of *delay_140_y* variable distribution. For this action, we transformed *delay_140_y* into a numeric datatype. The optimal binning for *living_area* was found to be 7 and the original variable was replaced by the binned version, see Figure 14. Finally, *delay_140_y* was transformed back to factor.

We calculated information value for the independent variables, with dependent variable being *delay_140_y* . We discovered that *mean_delay_1m*, *mean_delay_3m*, *mean_delay_6m*, *delay_indiv_140* and *delay_indiv* provided the most information value.

The dataset was then split into training (60%), validation (20%) and test (20%) datasets. We used stratified sampling to avoid missing classes in training data. Hypergrid was defined to tune the parameters. After fitting the model, variable importance was calculated and we discovered that *mean_delay_1m*, *contract_status6*, *contract_status8*, *delay_indiv_140* and *delay_indiv* provided the best results.
Afterwards, we used hold-out to determine cutoff and to find the best values for alpha and lambda using training data. The optimal cutoff value which maximized both specificity and sensitivity was found to be 15.1%. Using alpha found in the previous step, we used 5-fold cross-validation to find optimal value for lambda (using validation data).
The accuracy of created model is 96.65%, and the AUC is 0.974 see Figure 15. For confusion matrix, sensitivity and specificity, please refer to the table below.

```
    Confusion matrix
preds2    0      1
    0 256849  3524
    1  6194  23598

Sensitivity : 0.87007
Specificity : 0.97645
```

Maximum accuracy for both cross-validation and hold-out method was achieved with hyper parameters *alpha = 0* and *lambda = 0*. Analyzing the hypergrid, we discovered that maximum accuracy was connected to lambda always being 0 and alpha then becoming irrelevant and ranging from 0 to 1. The best results were achieved with no penalization. The top decile lift is 8.667, for lift curve please refer to 17.

3.3 Estimation of the expected number of days of delay when the client triggers first action

The goal of the last model is to estimate the average number of days delayed if the client exceeds the first action. The expectation of our sponsor was to create a model better than the Simple average model by at least 30%. We decided to use Elastic net regression for the predictions. Simpler model was selected because of little prior experience with data science among team members.
First step was to check the correlations of the attributes. As can be seen on Figure 1, higher positive correlation was discovered between *due_amount* and *paid_amount*, therefore we decided to exclude *due_amount*. We also excluded attribute *due_date*, because delay was calculated as the difference between *due_date* and *payment_date*.
In the next step, the dataset was filtered by the rows, where *delay_21_y* is equal to 1, therefore we selected only payments with delay greater than 21 days. Variable *delay_21_y* was then excluded. The data was shuffled and split into training (60%), validation (20%) and test (20%) datasets. We used stratified sampling to avoid missing classes in training data.
For the Simple average model, we computed the average of trainval data and used it as a prediction on test set. The average delay of payments that exceeded the first action was **128.2179** days. Simple average model provided root mean squared error (RMSE) **213.3069**.
For the Elastic net regression, hypergrid was defined to tune the parameters. We decided to use hold-out method as well as 10-fold cross-validation to find the optimal values of hyper parameters. Results can be found in the table below.

alpha	lambda	rmse_ho	rmse_cv
0	0	53.51514	53.16496

As can be seen, minimum RMSE of both cross-validation and hold-out was selected with hyper parameters *alpha = 0* and *lambda = 0*. Analyzing hypergrid, we discovered that minimum RMSE was connected to lambda being always 0 and alpha then becoming irrelevant and ranging from 0 to 1. The best results were achieved with no penalization. Our model exceeded the Simple average benchmark by 74%. Variable importance of the CV regression model can be seen on Figure 18.

4 Discussion

All three models passed the criteria desired by our sponsor. The AUC for model prediction delay of 21+ days was 0.946 and its accuracy 88.66%. The AUC for the second classification model was 0.974 and its accuracy was 96.65%. We assume such good results were achieved due to thorough feature engineering which captured clients' past behavior accurately and due to this past behavior being the most valuable knowledge for predicting future behavior.

The specificity of both the first and the second model exceeded the sensitivity, meaning that the proportion of payments correctly identified as not being delayed past 21 (140) days was higher compared to the proportion of payments correctly predicted as delayed past 21 (140) days. However, it is not known whether the actual cost reduction due to correctly identifying payments which will not be delayed is higher compared to the cost reduction due to correctly identifying payments which will be delayed past a specific threshold.

The third model predicted that the average delay for payments, which were delayed by more than 21 days, was 128.2179 days. The generated model was 74% better than simple average model. We suspect this could be due to the use of *mean_delay_1/3/6/12m*, as these features include simple arithmetic mean calculated before splitting the data into training, validation and test datasets. On the other hand, these varibles do not seem to carry high value for prediction, as evidenced in 18 and after excluding them the RMSE increased by approximately 10.

5 Conclusion

The goal of this data science project was to aid in optimization of the payment collection process by predicting if the clients delay their payments past 21 days and past 140 days. The third task focused on predicting the average number of days delayed if the client exceeds the first action (21 days).

We generated two classification models and one prediction model using existing customer data. The techniques used were penalized logistic regression for classification and regularized linear regression (elastic net) for prediction. The success criteria for classification models were set as AUC>0.7, the prediction model was expected to perform better than simple average model by at least 30%. All of our models managed to exceed these expectations.

As the collection process in case of client's delay originally consisted of three thresholds (21 days, 70 days, 140 days), we suggest creating a new model predicting if the client delays payment past 70 days. Additionally, conditional probabilities between the delays at each threshold should be explored, as each significant delay implies increased costs. As for the prediction model, we recommend deriving *mean_delay_1/3/6/12m* features after splitting the data into training, validation and test datasets. We further recommend exploring mean delay in days for delays past the second and the third threshold. In addition, we recommend to explore more sophisticated methods, such as neural nets for classification and regression trees or neural nets for prediction. From business perspective, it might prove useful to perform cost-benefit analysis based on confusion matrices from the classification tasks (identifying costs associated with true positives, true negatives, false positives and false negatives).

6 Figures

7 Tables

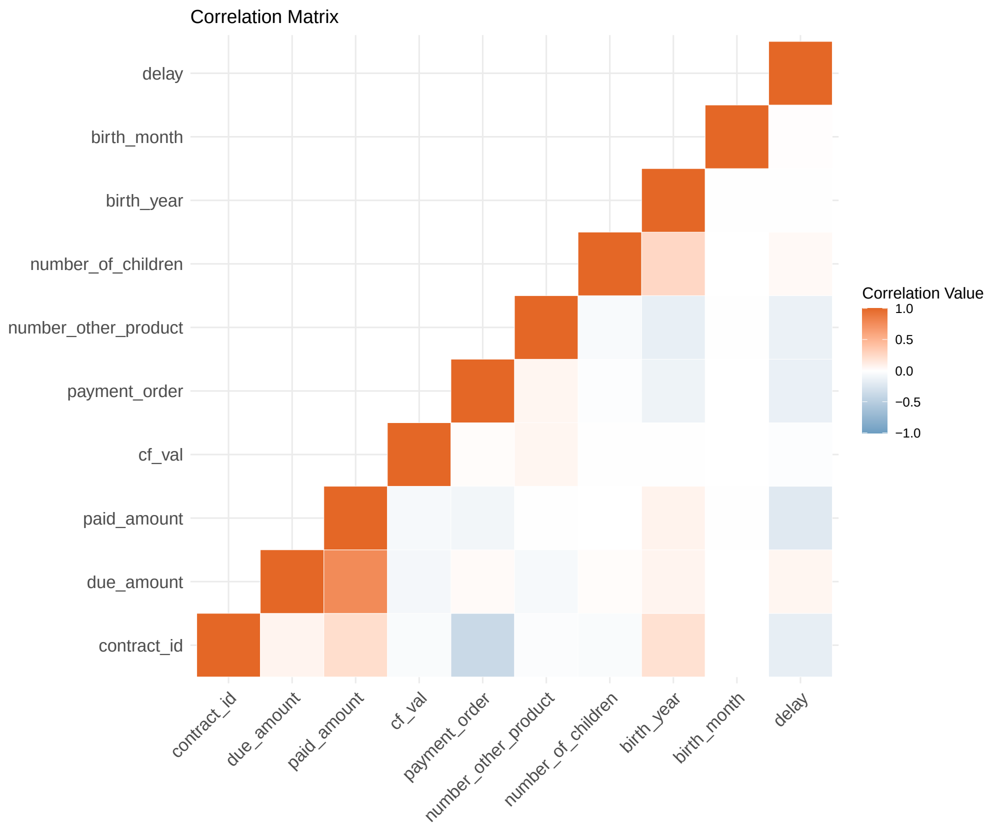

Figure 1: Correlation plot.

Table 4: Statistics of missing values.

variable	n_miss	pct_miss
different_contact_area	471354	20.0319420
cf_val	469310	19.9450747
living_area	469015	19.9325375
kc_flag	468906	19.9279052
payment_order	83361	3.5427359
payment_date	11733	0.4986375
delay	11733	0.4986375

Table 5: Statistics of missing values of the new variables.

variable	n_miss	pct_miss
mean_delay_12m	804268	55.435142
mean_delay_6m	458935	31.632648
mean_delay_3m	249422	17.191712
mean_delay_1m	95489	6.581694

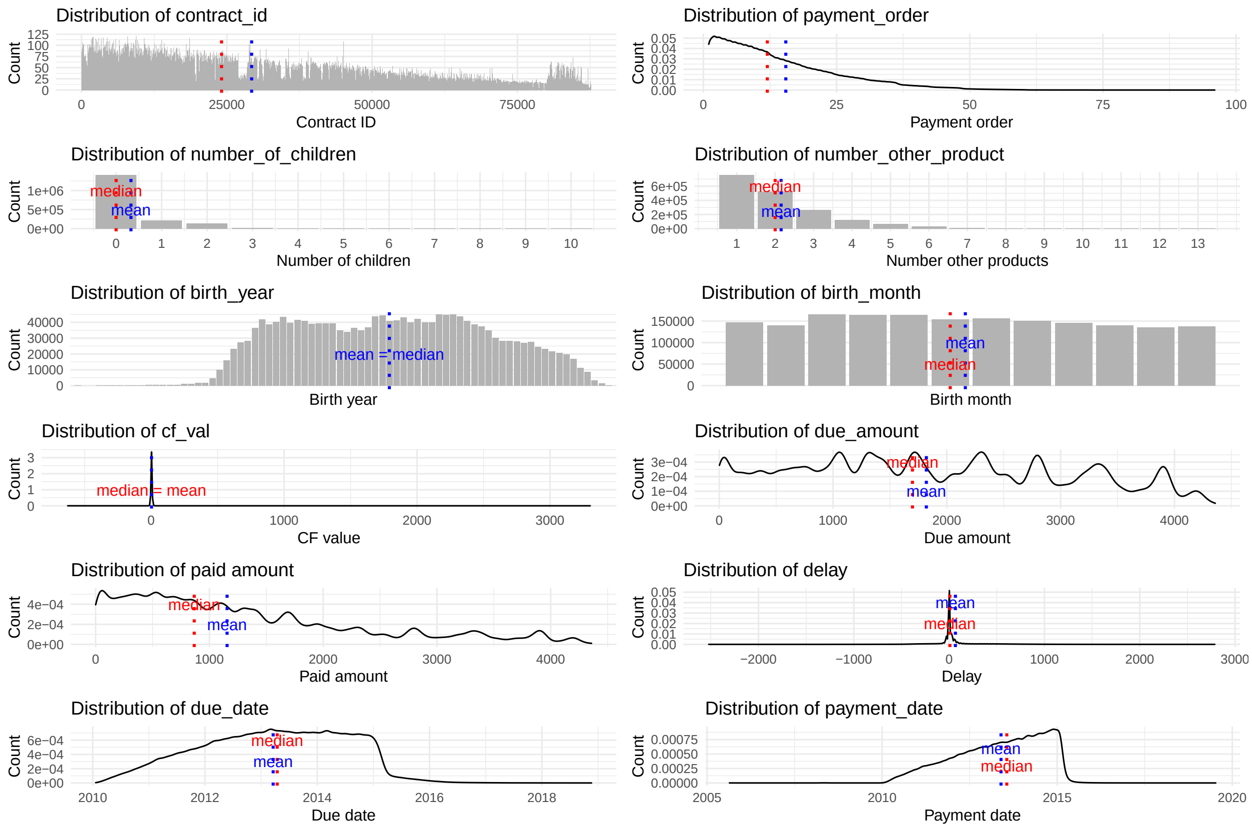

Figure 2: Density plots.

Table 6: Statistics summary.

headofTable	EX	VarX	Median	Q1	Q3	Min	Max
Num. of Children	0.3257497	5.010432e-01	0	0	0	0	10
Num. Other Product	2.1550566	1.981590e+00	2	1	3	1	13
Year of Birth	1964.9877404	1.884125e+02	1965	1953	1976	1921	1996
Due amount	1820.2133351	1.327569e+06	1698	876	2754	2	4360
Paid amount	1155.5253292	1.006110e+06	1704	382	1651	2	4360
Delay	22.1283613	4.693719e+04	NA	-2	34	NA	2787

Table 7: Statistics summary of the new variables.

headofTable_new	EX_new	VarX_new	Median_new	Q1_new	Q3_new	Min_new	Max_new
delay_indiv	25.679340	10196.8875	11.750000	-0.4000000	31.31579	-1673.000	1980.000
delay_indiv_21	7.068964	101.1708	2.000000	0.0000000	11.00000	0.000	60.000
delay_indiv_140	0.819534	9.7183	0.000000	0.0000000	0.00000	0.000	44.000
mean_delay_1m	19.642073	40286.2966	10.000000	-1.0000000	32.00000	-1673.000	2068.000
mean_delay_3m	16.733327	33096.2809	4.666667	-0.3333333	31.33333	-1398.000	1715.000
mean_delay_6m	12.864224	25225.8353	0.000000	0.0000000	30.16667	-1352.167	1597.500
mean_delay_12m	6.883395	14643.2779	0.000000	0.0000000	22.00000	-1260.333	1489.917

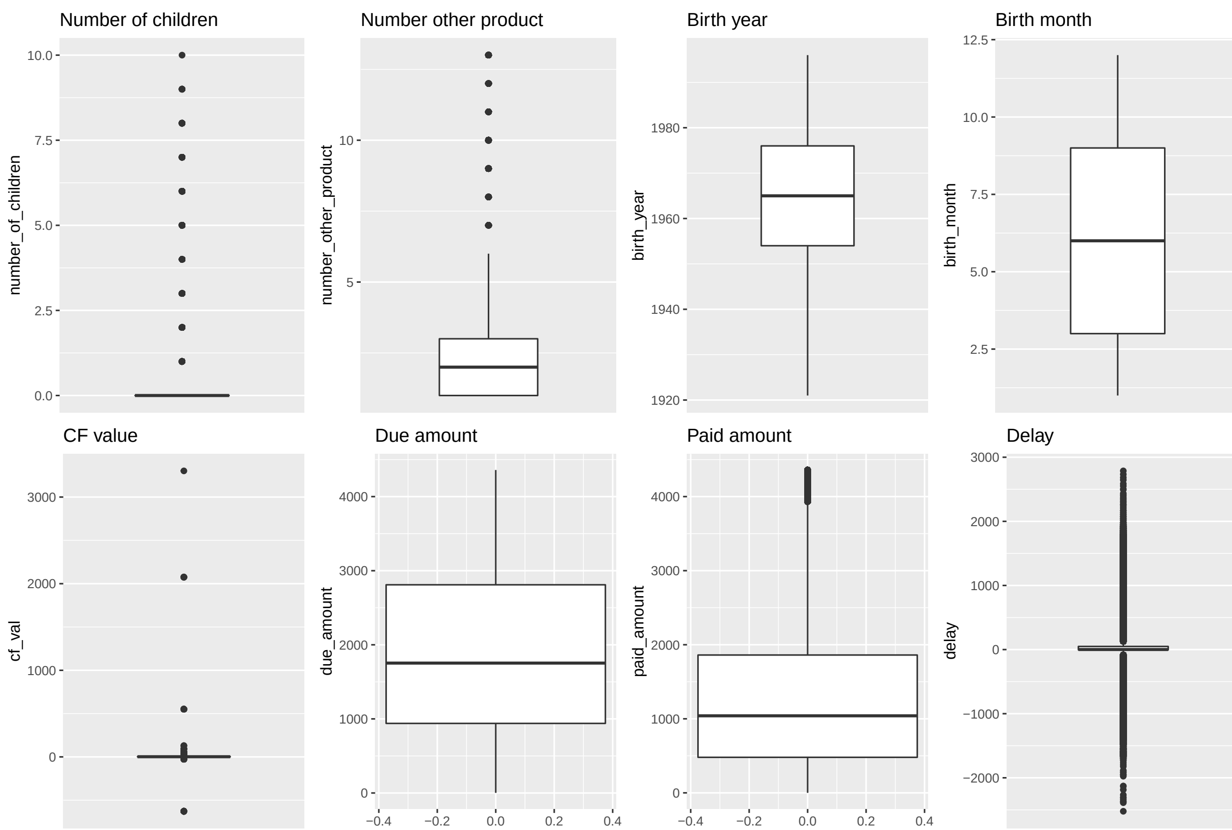

Figure 3: Boxplots for numeric attributes.

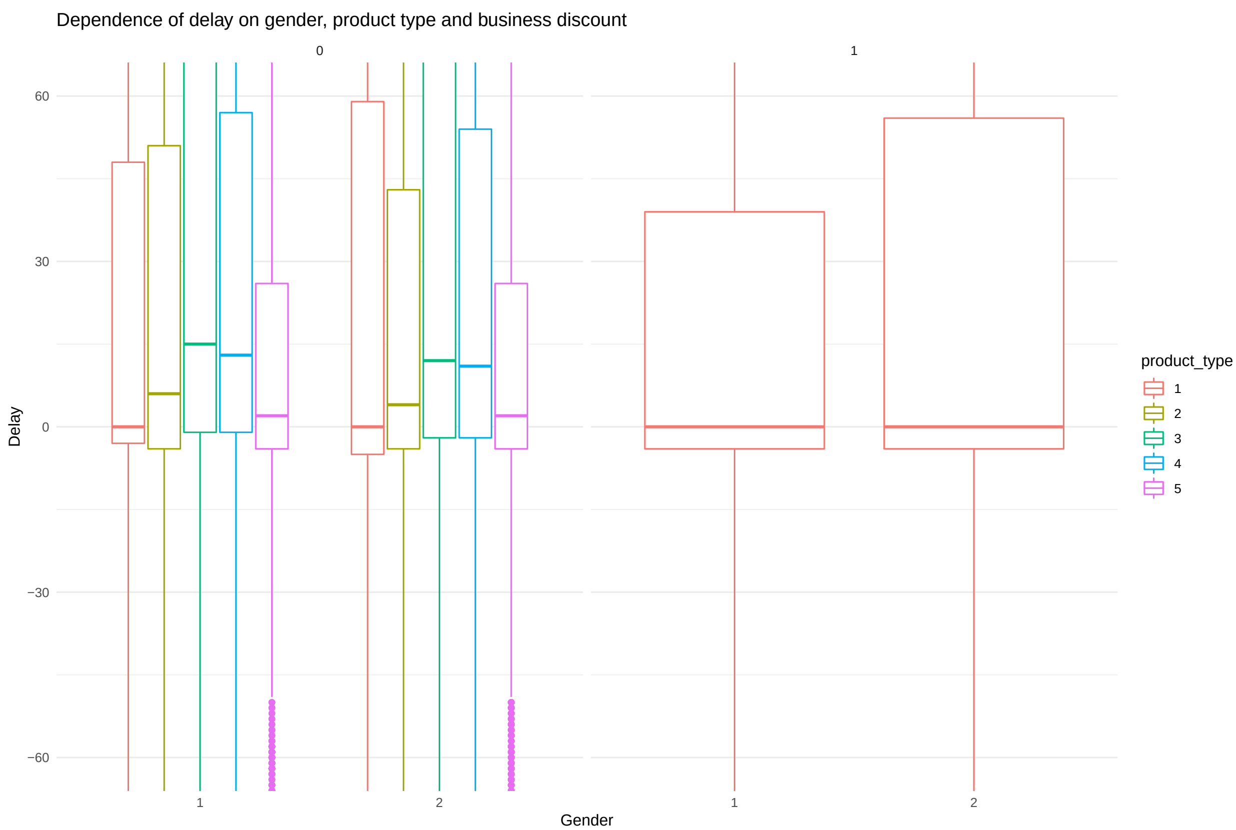

Figure 4: Dependence of delay on gender, product type and business discount.

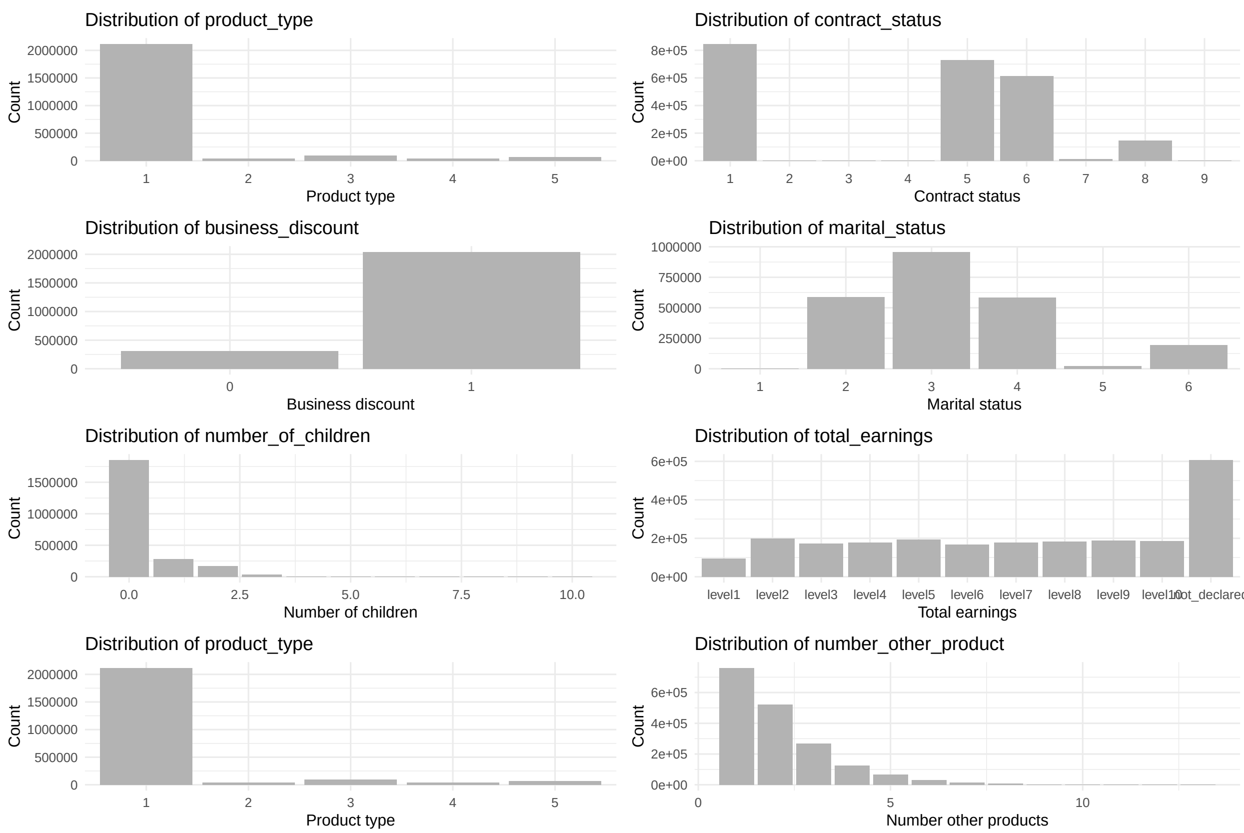

Figure 5: Distribution plots.

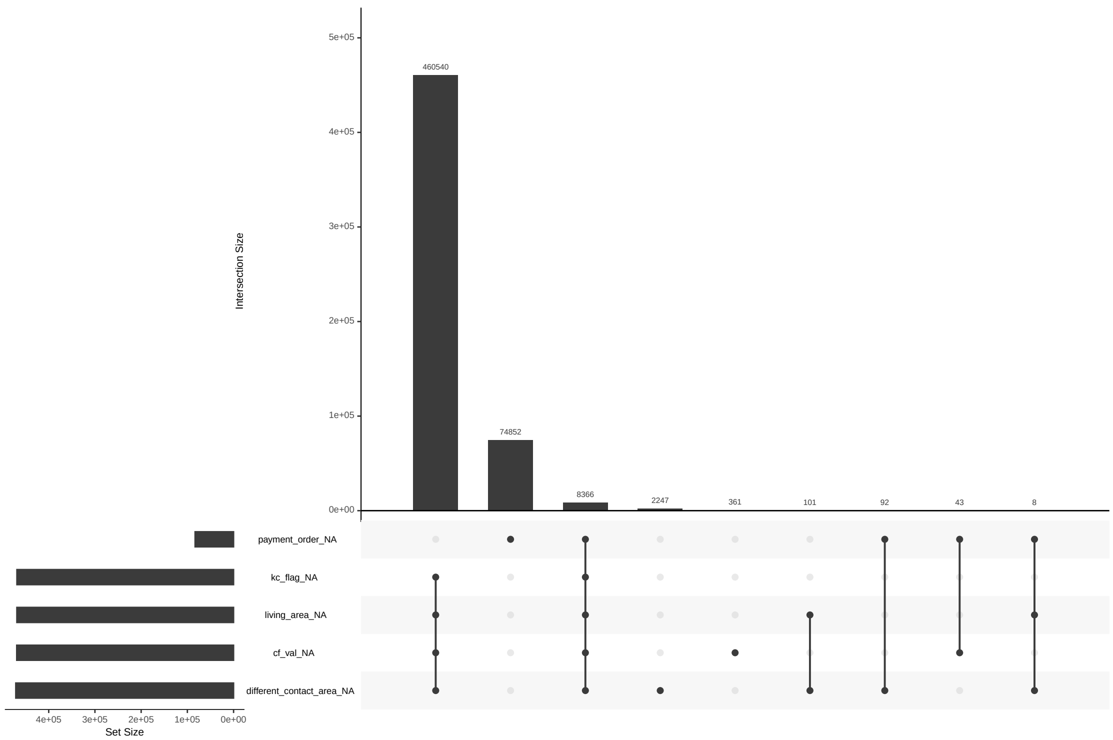

Figure 6: Distribution of missing values.

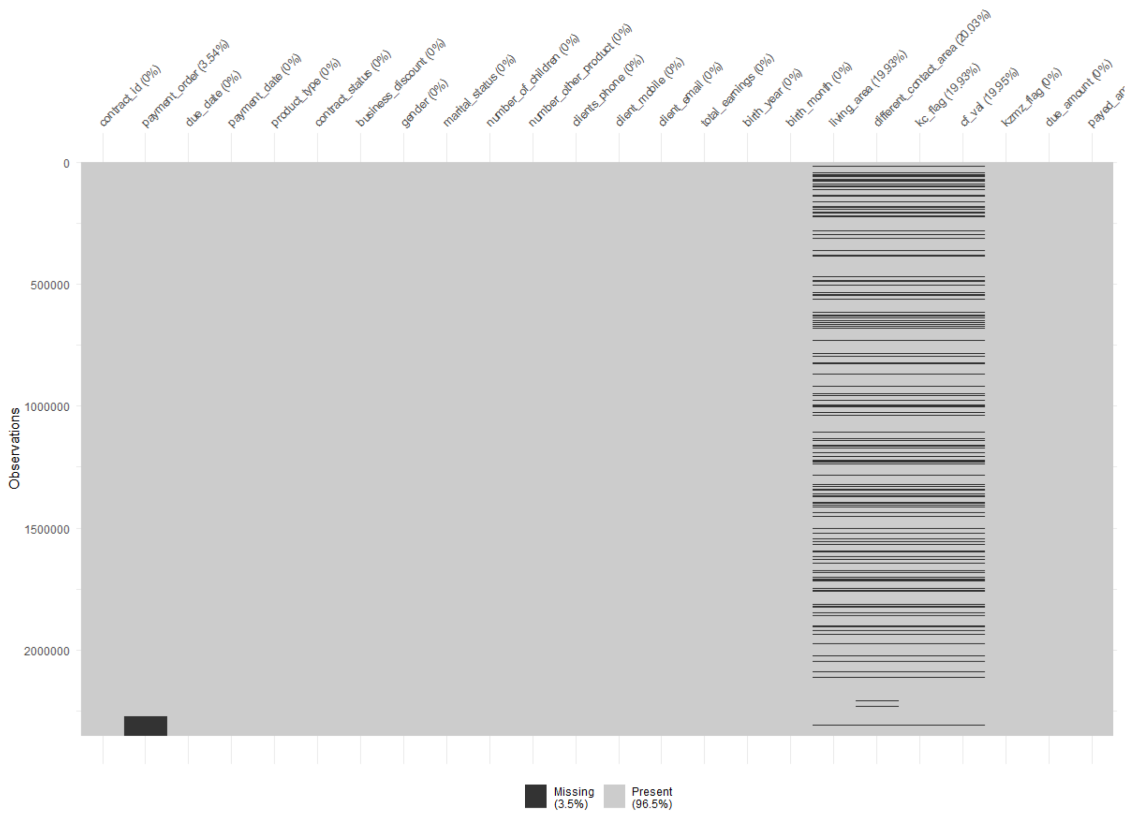

Figure 7: Distribution of missing values

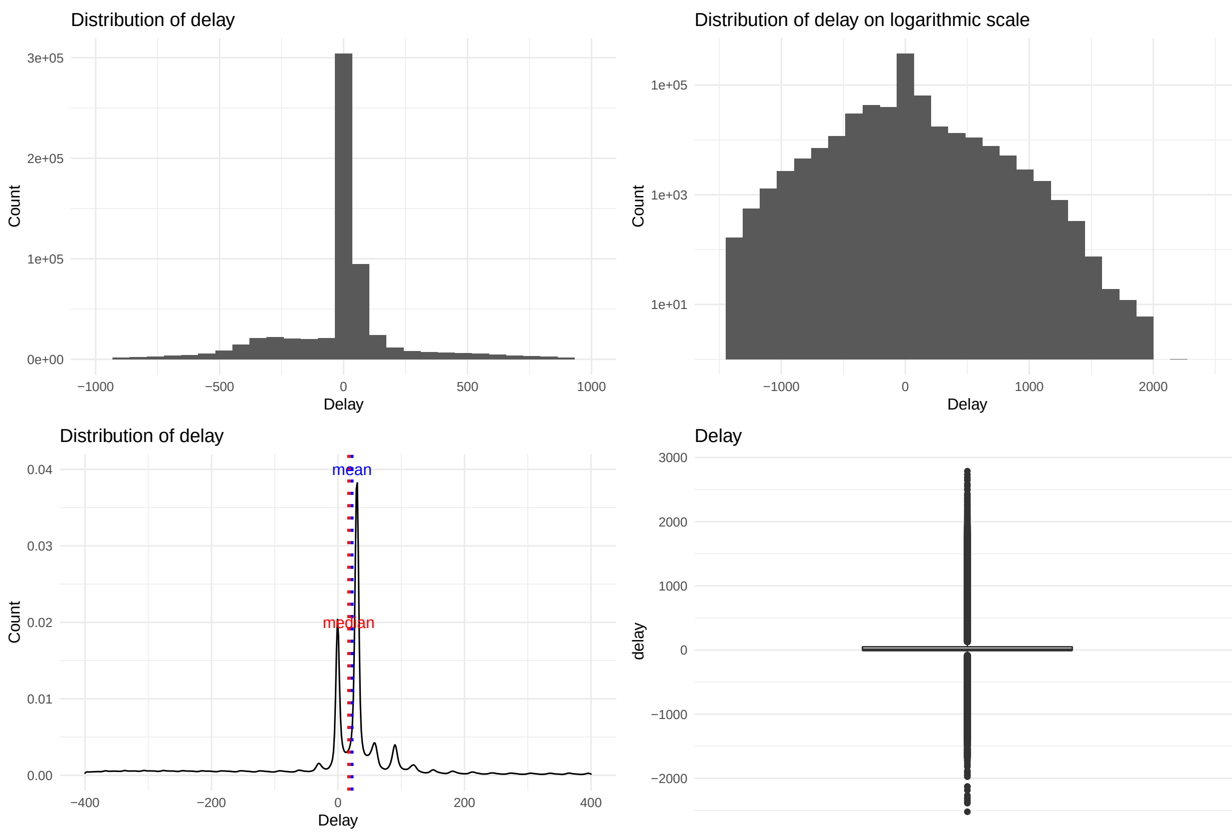

Figure 8: Basic statistics of the added attribute delay.

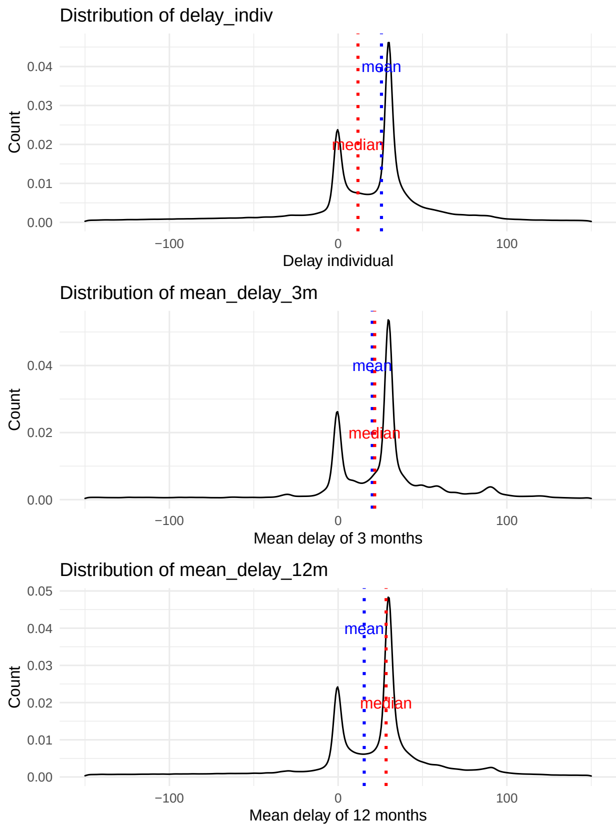

Figure 9: Basic statistics of the added attributes.

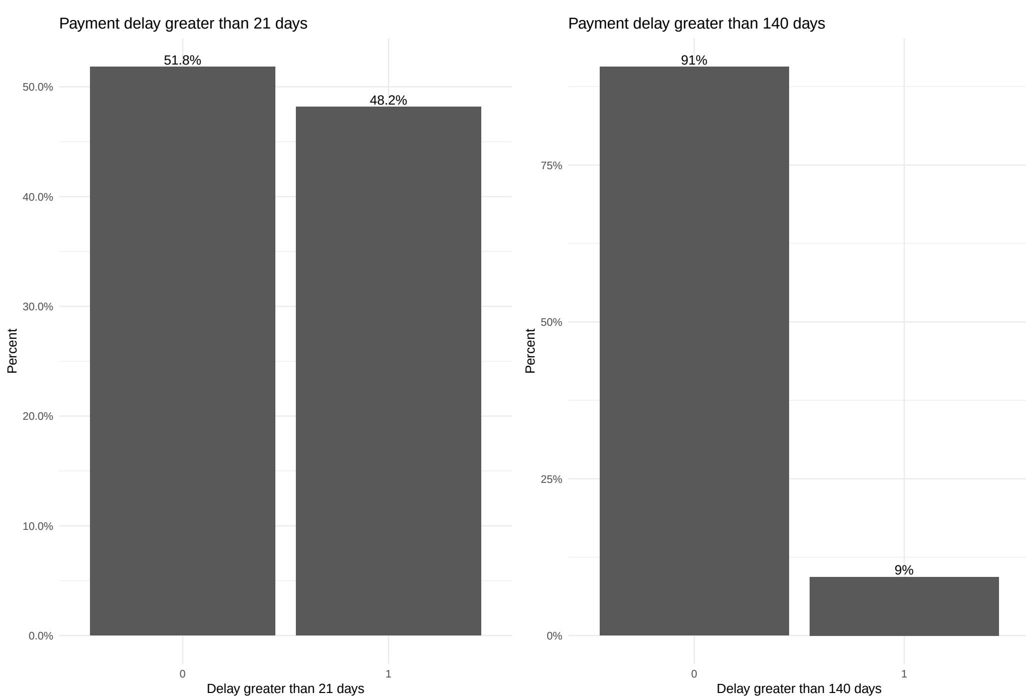

Figure 10: Statistics of the factor added attributes.

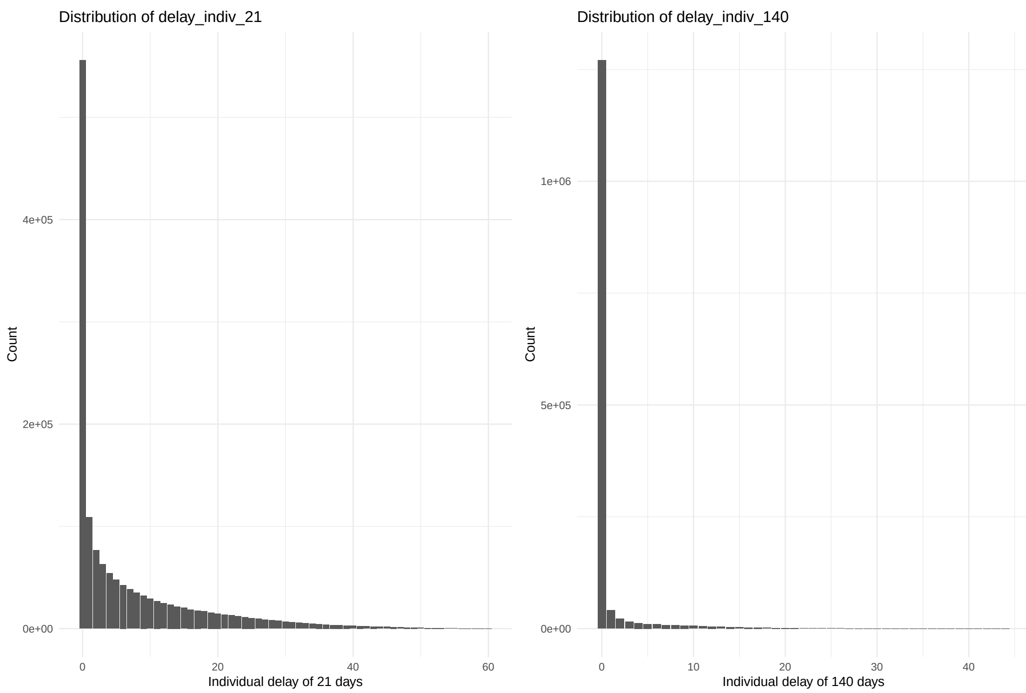

Figure 11: Basic statistics of the added attributes.

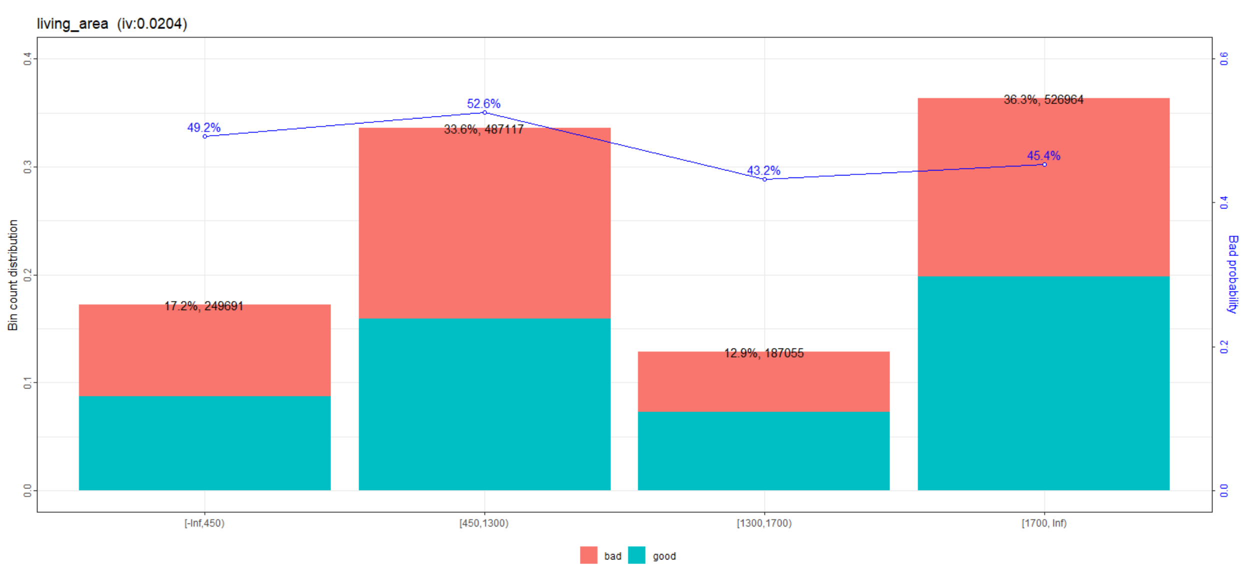

Figure 12: Binning for living_area (21+ delay)

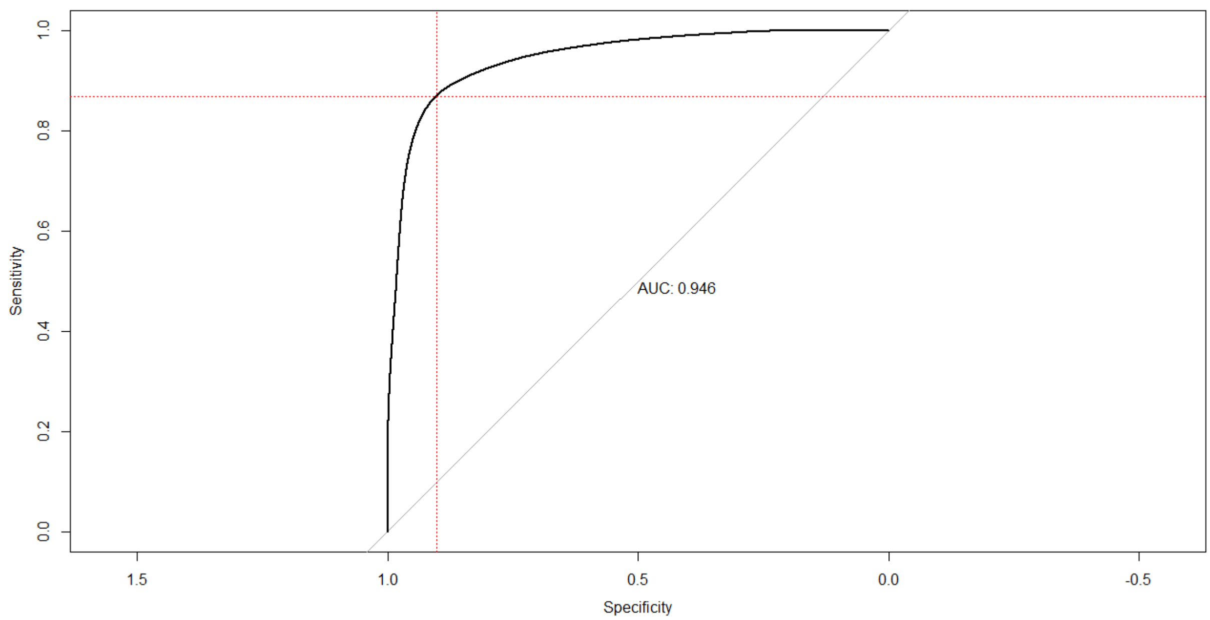

Figure 13: ROC curve and AUC (21+ delay)

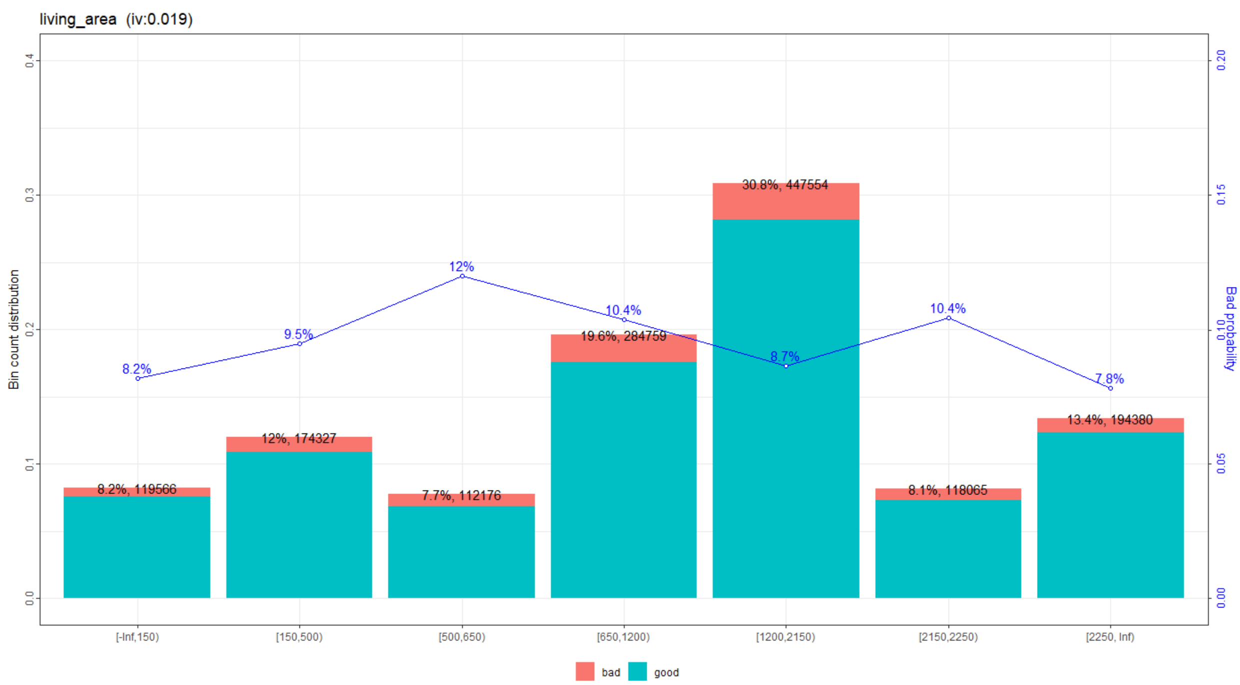

Figure 14: Binning for living_area (140+ delay)

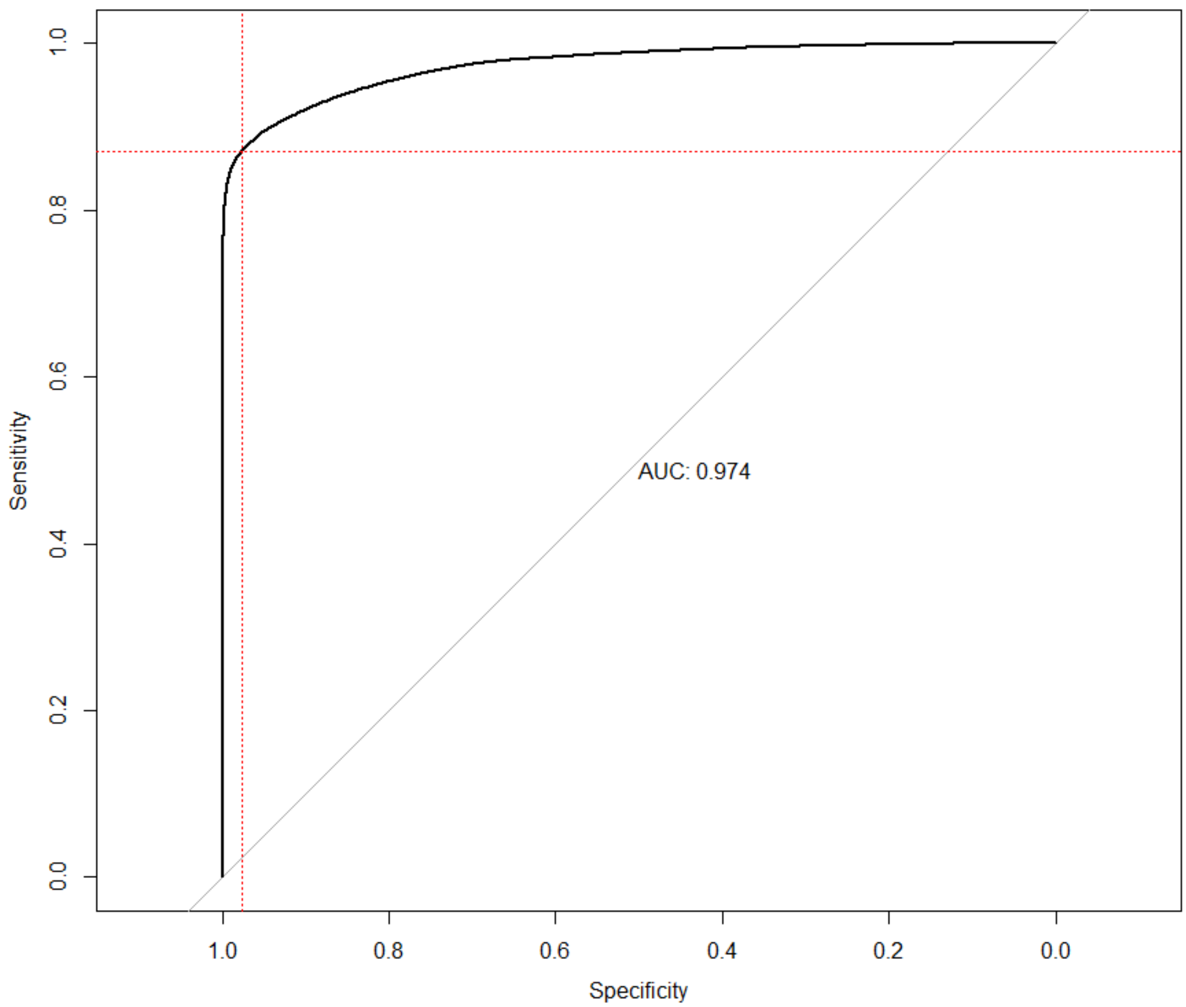

Figure 15: ROC curve and AUC (140+ delay)

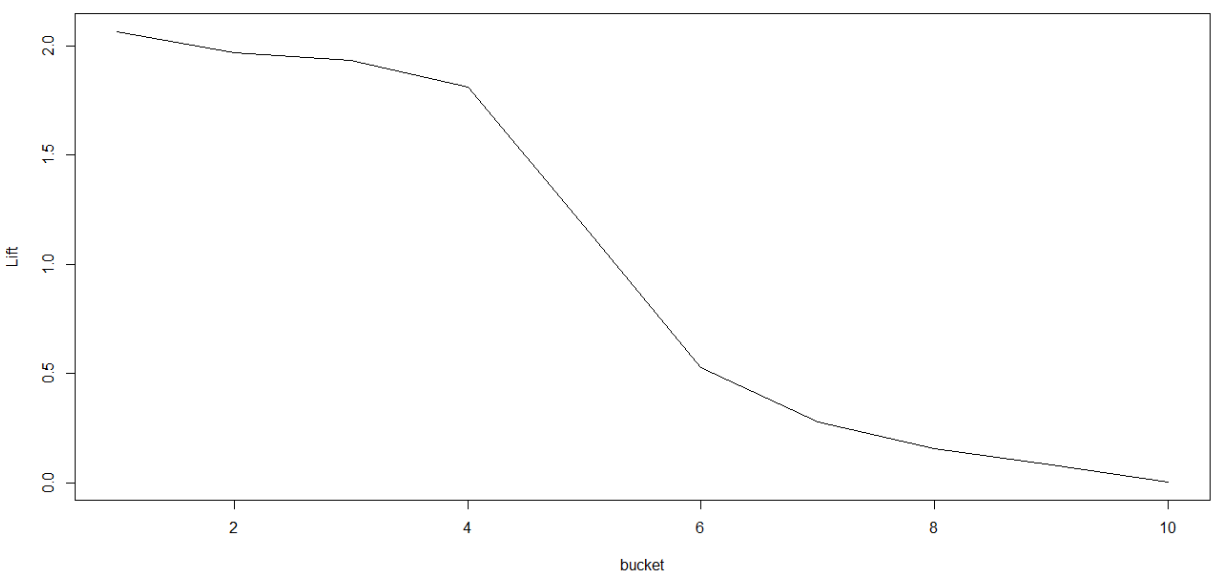

Figure 16: Lift curve (21+ delay)

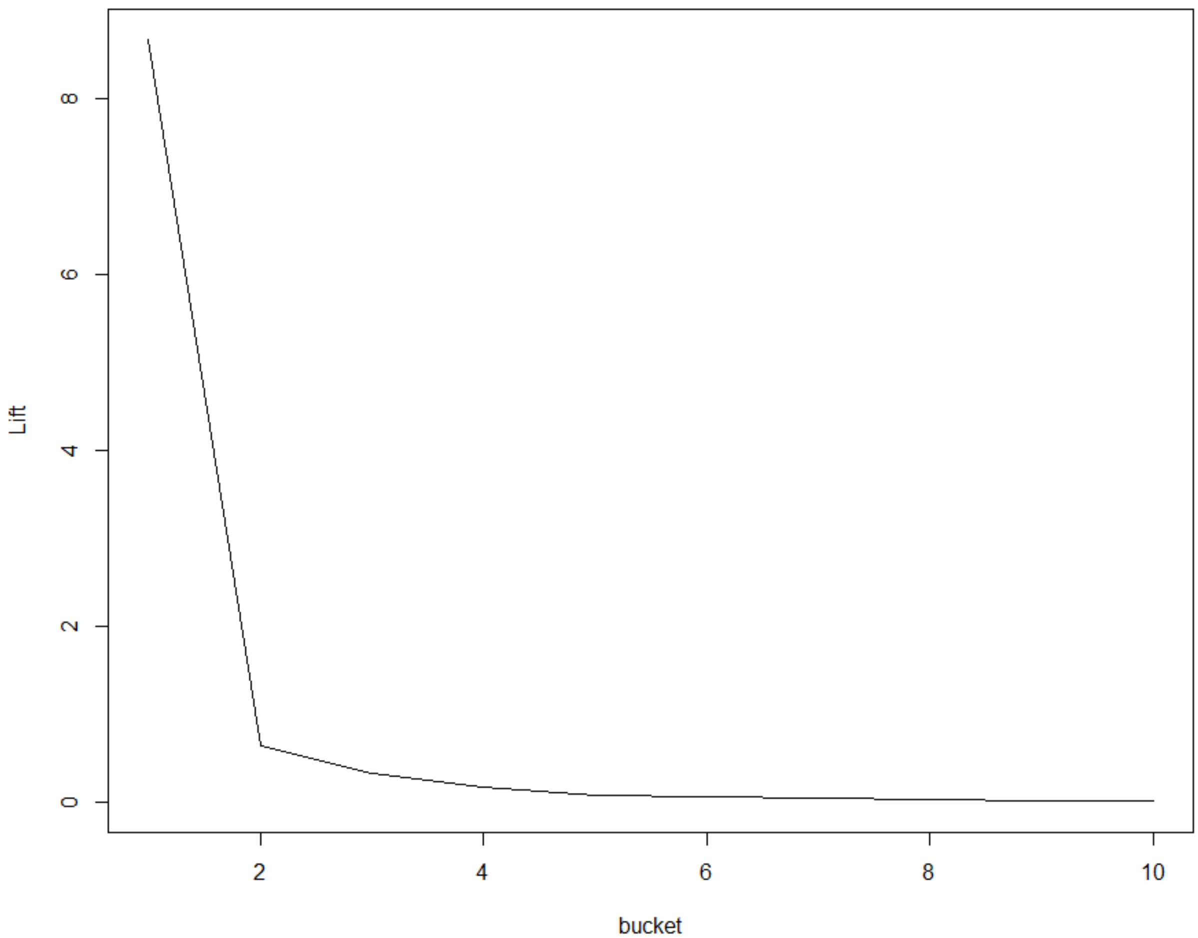

Figure 17: Lift curve (140+ delay)

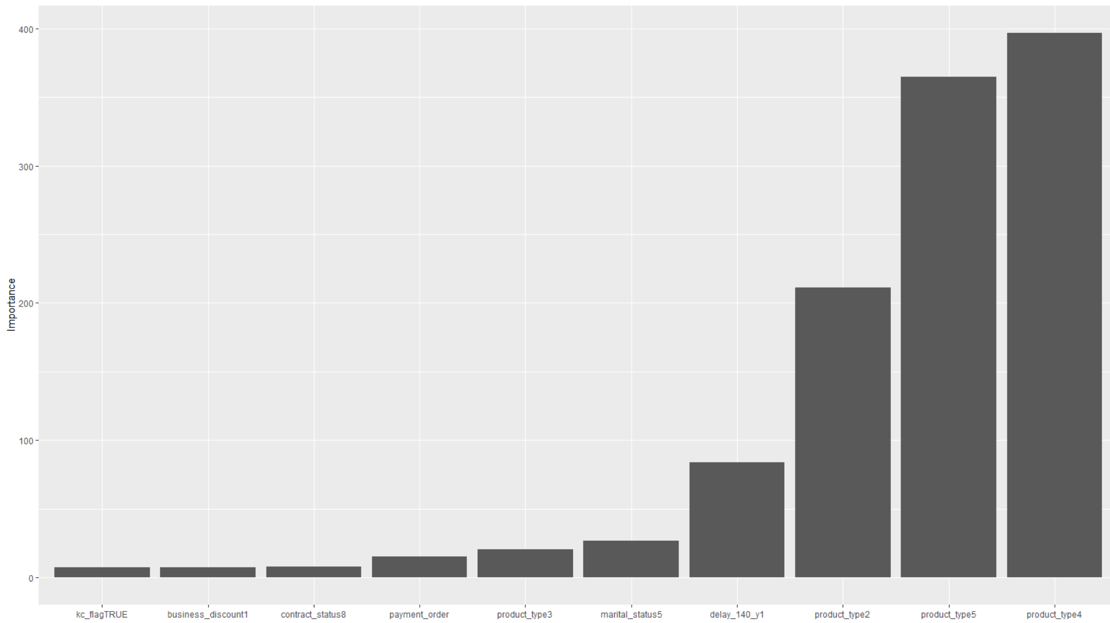

Figure 18: Variable importance for glmnet cv

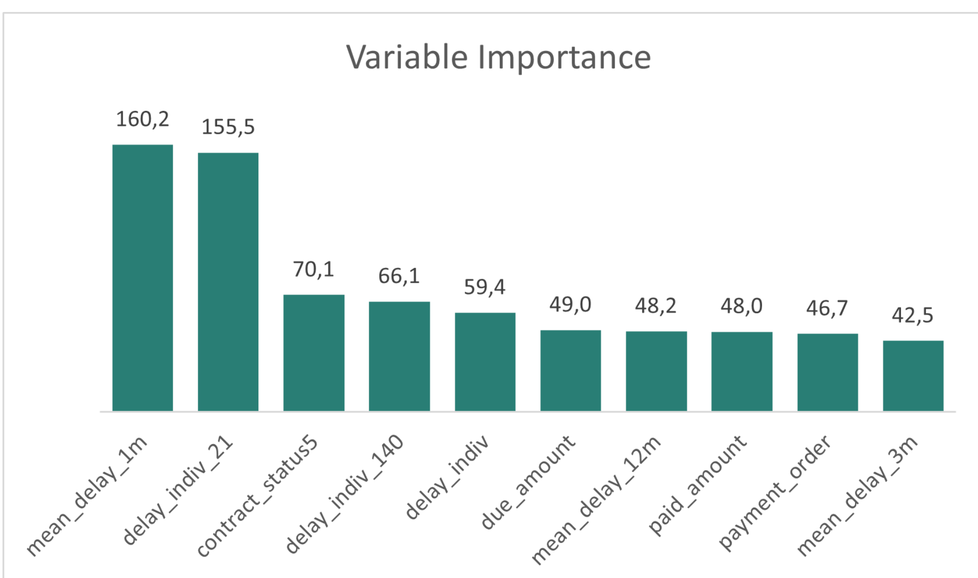

Figure 19: Variable importance (delay 21+)

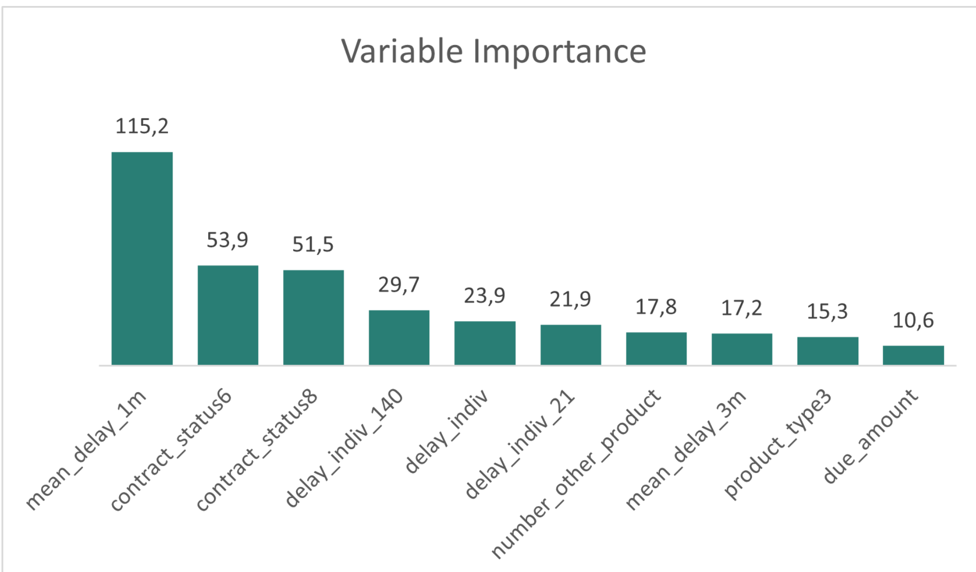

Figure 20: Variable importance (delay 140+)