MICRO CREDIT PREDICTION

Submitted by:

KONATALA MOHIT

ACKNOWLEDGMENT

I am grateful to the writers, developers and authority of the website pages machinelearningmastry, geekforgeeks, wikipedia and stackoverflow which have helped me to refer for whenever there was need for guidance. Articles on Indonesian Telecom sector, Bing networks and Economics in finance helped me to get a better understanding of the data to analyse and explore data. I am mostly grateful to DataTrained who have guided me in learning and enhancing my data science skills required to analyse and solve this project.

INTRODUCTION:

In this report I will be discussing about micro credit prediction using few of the machine learning models via python and its libraries. In simple terms micro credit means a loan of small amount. The primary objective of micro credit is to help raise the income and standard of living of the below poverty line category families. Micro credit is only applicable to lower income families and it is provided by micro credit institutes of the country. Communication devices have become primary digital gadget to pursue our normal life in today's life. On an average a person uses his phone for 4 hours of the day i.e., texting, internet surfing, streaming online or calling etc. Telecom industries have also initiated micro credit service to help provide services to lower income class which would help sustain their lives.

The drawback of micro credit services is the uncertainty of the subscriber to payback the amount allotted to them within 5 days of time. To get the better idea of this we will look at the various factors of data provided by one of the clients to showcase if the telecom subscriber can return borrowed loan amount within the given time period.

Review of Literature:

For the better understanding of the data referred to telecom sector of Indonesia as the data is of that particular country and micro credit in finance. Micro credit was originated from Grameen Bank that was developed by economist Muhammad Yunas i.e., established in 1976 in Bangladesh. Telecom micro credit is a value-added service particularly designed to meet consumer needs. It is a service which the telecom can offer to subscribers and resellers both. Through this service, telco can provide ease of access of airtime stock credit to their resellers when they run out of stock balance, without disrupting the normal flow of business. On the subscriber side, they can remain connected without having to recharge through a voucher or digital topup. This loan is disbursed in real-time and is calculated based on intelligent reseller/subscriber profiling.

Undertaken Problem:

Objective of the project is to predict if the loan borrowed by the customer can be cleared by that person with a span of 5 days or not i.e., by analysing various factors that affect the label outcome. Micro Credit prediction is Classification type prediction as the label is either yes or no.

Mathematical/ Analytical Modelling of the Problem:

Label is the target data in the dataset and the variables present the target data are in binary form, therefore Classification Machine Learning Models are used in this project. Classification is for estimating categorical form of data by using established relation

between feature and target variables. Descriptive Analysis is used to study and observe the data.

Data Sources and their formats:

The source of the data is provided by the client. The data is provided in the csv format. Data contains 209593 entries having 37 variables in the dataset.

Data Pre-processing:

Data pre-processing has two main steps i.e., Data Cleaning and Data Transforming.

Data Cleaning:

Data Cleaning is one of the most important steps creating a machine model. If an uncleaned data is fed to a machine learning model, then the model will perform very poorly.

The Dataset does not contain any null values in it. Unnamed: 0 (serial number), msisdn (mobile number) these two columns have unique element for every individual row therefore dropping them. Pcircle column has only one unique element and it is repeated in every row so it won't help us the outcome prediction therefore dropping this variable. Pdate columns can be simplified by splitting it into day, month and year columns by using split method and assigning values to new columns. Removing the pdate columns as it has been divided into 3 new columns and also removing year column as the data is of only of year 2016. Day and month are in object format converting them into numerical format using pd.numeric.

Few of the columns contains negative values which is can't possible so locating those negative values and multiplying them with -1 to make them as positive values. The dataset contains outliers. Age of the cellular maximum value is 999860 days which is inappropriate in general sense. So, limiting the data to 1465 days as considering the client is a new telecom industry launched 2012. In Indonesia the average monthly expenditure per capita in mobile bill in 2016 is 22182 Indonesian Rupiah. According, to that per day daily money spent can be around 740 (maximum). Therefore, limiting the maximum value of daily value to 740 for both 30 and 90 days. We know that 22182 is the maximum value and loan is used by the lower income families so the upper limit plan of lower income used is around 7500 so any value around that is rounded to 7500 as monthly recharge and for 90 days it is 10000 Indonesian Rupiah. Last recharge date has been limited to 30 days as most of the package used by lower income class is monthly plan. Recharge frequency has been limited to 1 as most of the people try to recharge only once a month unless it is for additional data. The max loan amount values have only 3 variety 0, 5 and 10 i.e., values been 1-4 as marked as 5 and any other value above is marked as 10. For remaining columns outliers can be handled by using IQR method.

Note: Label column is imbalanced.

Data Transforming:

After Data Cleaning the data must be transformed into numerical form as one can't feed ordinal data to machine learning model and also the data must be normalized as normalizing the data the machine learning model gives equal importance to all the data.

In this project I am using label encoder to transform the ordinal data present in the data set into numerical form. Using Standard Scalar function to normalize the data.

Note: Before normalize the data separated the target variable from feature variables.

Using VIF to check that value is under 5 as having higher number indicates that the dataset the multi collinearity between the independent variables which must be arrested else we cannot achieve optimal machine learning model.

After standardizing the data, we must split the data into train and test sets. Train Test Split method is to split the data into train and test data

Data Inputs- Logic- Output Relationships:

After Separating the Input (feature) and output (target) data we split them into train and test division one part of the data is used to train the ML model and other part of the data to predict the output. After splitting the data, we must balance the training data in order to minimize the ML being biased the majority result. Using SMOTE to balance the training data. When the train data is fed to machine learning model it generates an algorithm or simply put an equation that is applicable to all the data and when test data is fed to it implements the trained data equation to the current input values to predict the outcome. If the input has no outliers and is clean that there is no over or underfitting in the outcome.

Hardware and Software Requirements and Tools Used:

Hardware Required for Jupyter Notebook Software is as follows:

Memory and disk space required per user: 1GB RAM + 1GB of disk + .5 CPU core.

Server overhead: 2-4GB or 10% system overhead (whatever is larger), .5 CPU cores, 10GB disk space.

Port requirements: Port 8000 plus 5 unique, random ports per notebook

Libraries Used:

Pandas library: To frame raw data, visualize and perform task on it via other libraries.

Numpy library: To perform mathematical functions on the framed data numpy is used. In this project used it to location nan values and replace them with desired value also to find mean and standard values.

Matplotlib library: This library is used to visualize the data. Used to visualize univariate and bivariate analysis (pie plot, count plot and scatter plot) also to visualize outliers via box plot.

Seaborn library: heatmap to see co-relation between feature variable to arrest high collinearity.

Sklearn library: Imported this library to normalize the data, split the data into train and test data, various machine learning model and cross validation techniques.

Warnings library: To ignore filter warning shown while compiling block of codes

Pickle: To save the trained machine learning model.

Algorithm Used:

1. XGB Classifier
2. Random Forest Classifier
3. K-Neighbors Classifier
4. Logistic Regression

Model Training and Selection:

XGB Classifier:

XGBoost stands for "Extreme Gradient Boosting" and it is an implementation of gradient boosting trees algorithm. The XGBoost is a popular supervised machine learning model with characteristics like computation speed, parallelization, and performance

```
model3 =XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.4,
              enable_categorical=False, gamma=0, gpu_id=-1,
              importance_type=None, interaction_constraints='',
              learning_rate=0.2, max_delta_step=0, max_depth=16,
              min_child_weight=4, monotone_constraints='()',
              n_estimators=350, n_jobs=8, num_parallel_tree=1, predictor='auto',
              random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1,
              subsample=0.6, tree_method='exact', validate_parameters=1,
              verbosity=None)
model3.fit(X_over,Y_over)
p3=model3.predict(X_test)
print(classification_report(p3, y_test))
```

```
[12:20:09] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/
0, the default evaluation metric used with the objective 'binary:logistic' was cha
t eval_metric if you'd like to restore the old behavior.
              precision    recall  f1-score   support

           0       0.51      0.56      0.53      4833
           1       0.94      0.93      0.94     37086

    accuracy                           0.89     41919
   macro avg       0.73      0.74      0.73     41919
weighted avg       0.89      0.89      0.89     41919
```

Random Forest Classifier:

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting

```
model4 =RandomForestClassifier(max_depth=16, max_features=10, max_leaf_nodes=50,
                       max_samples=0.3, min_samples_leaf=300,
                       min_samples_split=4, n_estimators=300)
model4.fit(X_over,Y_over)
p4=model4.predict(X_test)
print(classification_report(p4, y_test))
```

```
              precision    recall  f1-score   support

           0       0.71      0.39      0.51      9523
           1       0.84      0.95      0.89     32396

    accuracy                           0.83     41919
   macro avg       0.78      0.67      0.70     41919
weighted avg       0.81      0.83      0.81     41919
```

K-Neighbors Classifier:

KNN is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same neighbourhood

```
model2 = KNeighborsClassifier(leaf_size=40, n_neighbors=1, weights='distance')
model2.fit(X_over,Y_over)
p2=model2.predict(X_test)
print(classification_report(p2, y_test))
```

```
              precision    recall  f1-score   support

           0       0.50      0.35      0.41      7505
           1       0.87      0.92      0.89     34414

    accuracy                           0.82     41919
   macro avg       0.69      0.64      0.65     41919
weighted avg       0.80      0.82      0.81     41919
```

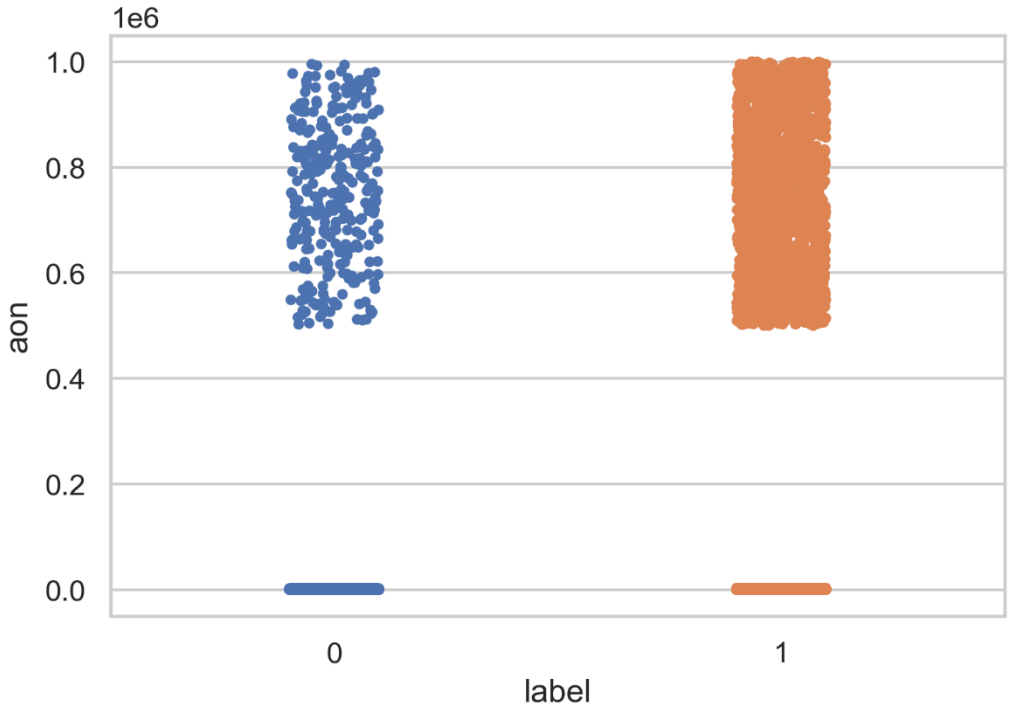
Logistic Regression:

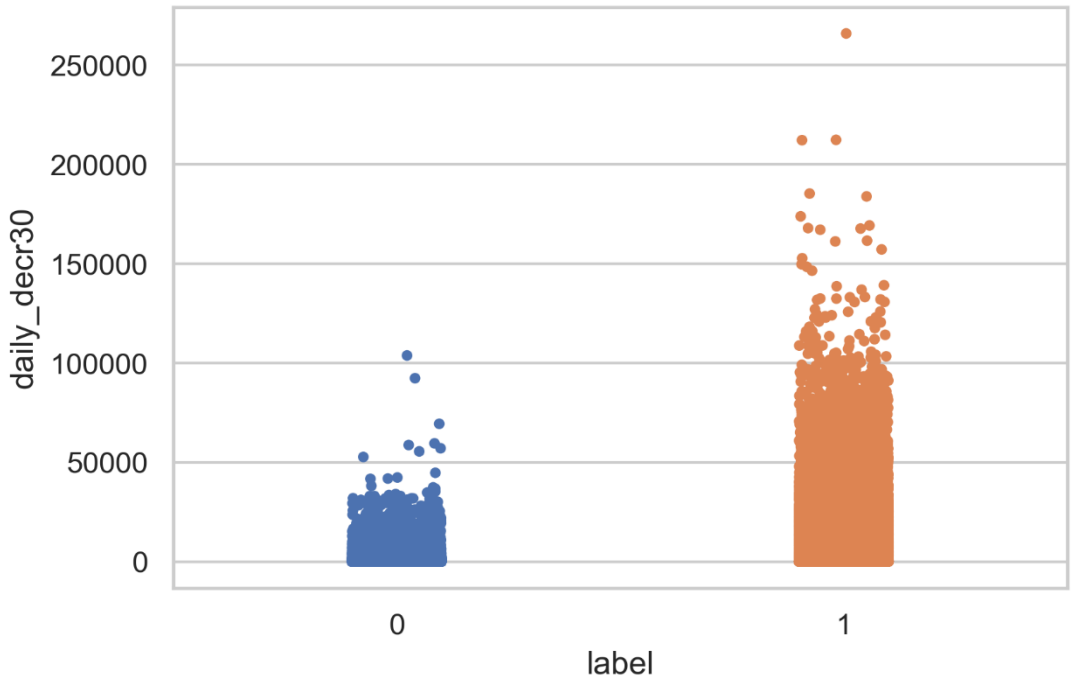
Logistic regression is a process of modelling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome, something that can take two values such as true/false, yes/no, and so on.

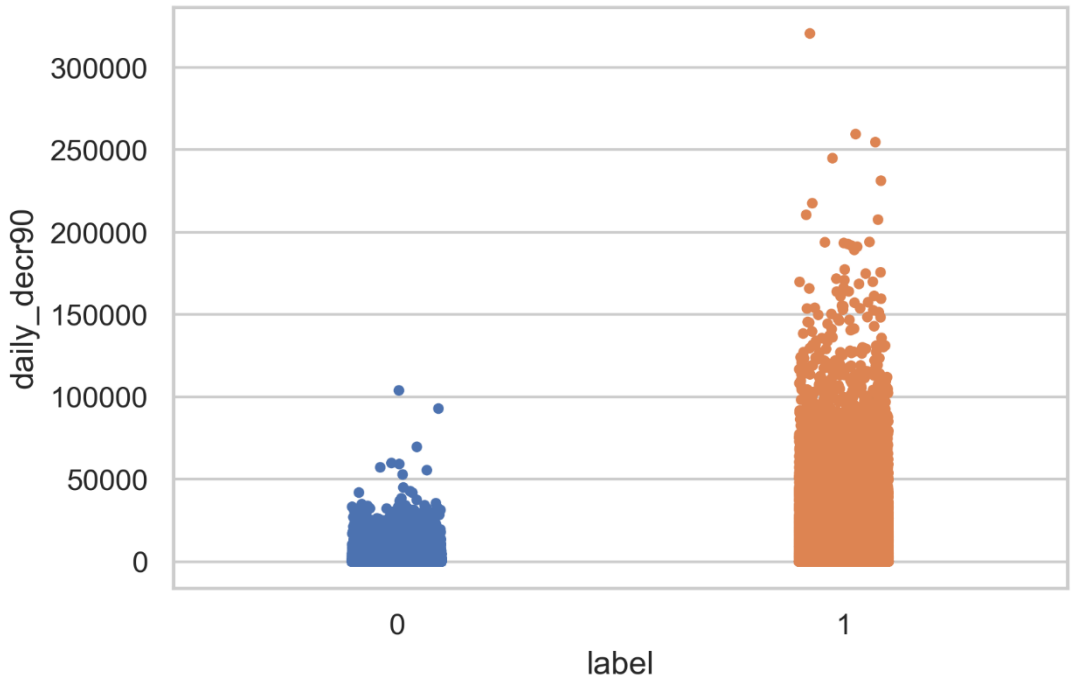
```
model1 = LogisticRegression(solver='newton-cg')
model1.fit(X_over,Y_over)
p1=model1.predict(X_test)
print(classification_report(p1, y_test))
```

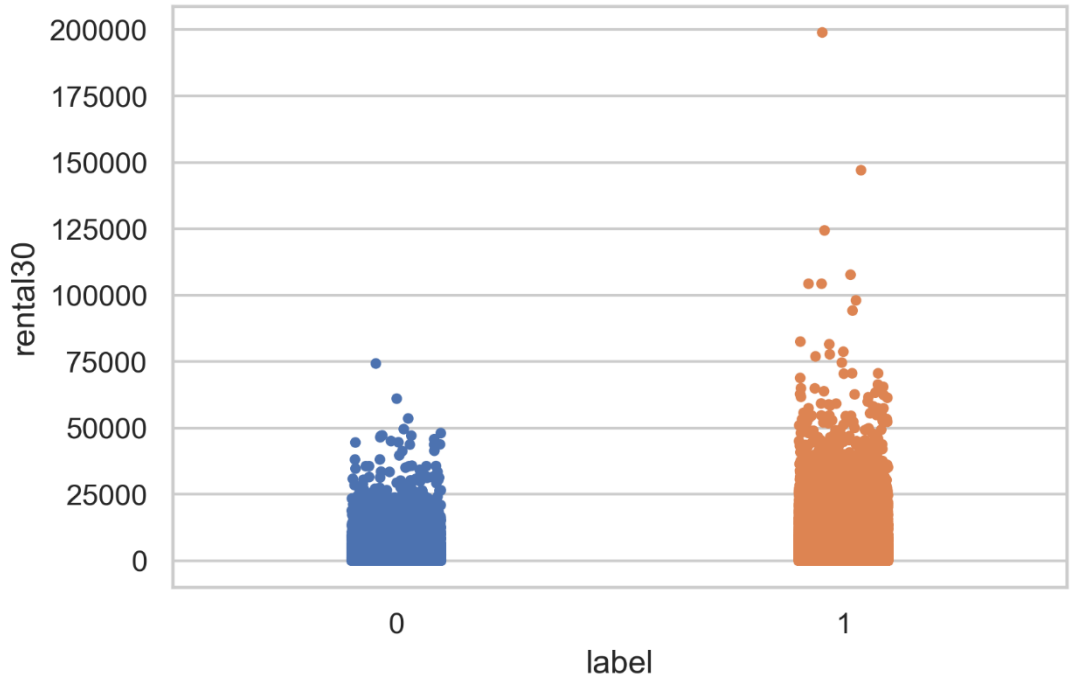
```
              precision    recall  f1-score   support

           0       0.77      0.28      0.42     14191
           1       0.72      0.96      0.82     27728

    accuracy                           0.73     41919
   macro avg       0.75      0.62      0.62     41919
weighted avg       0.74      0.73      0.69     41919
```

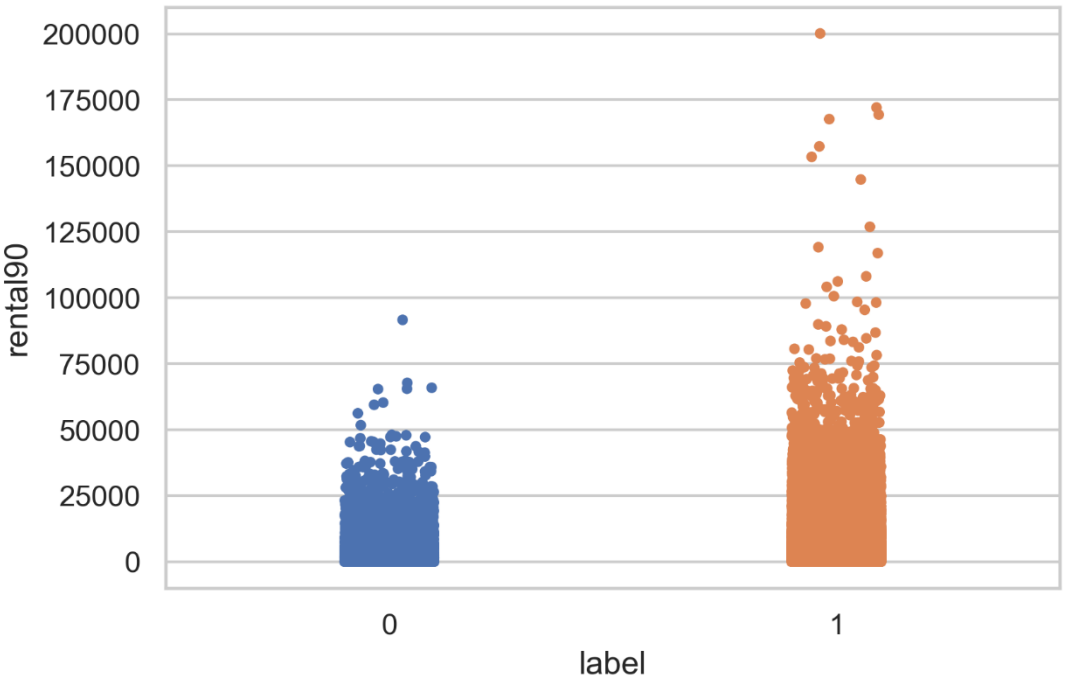
Visualization:

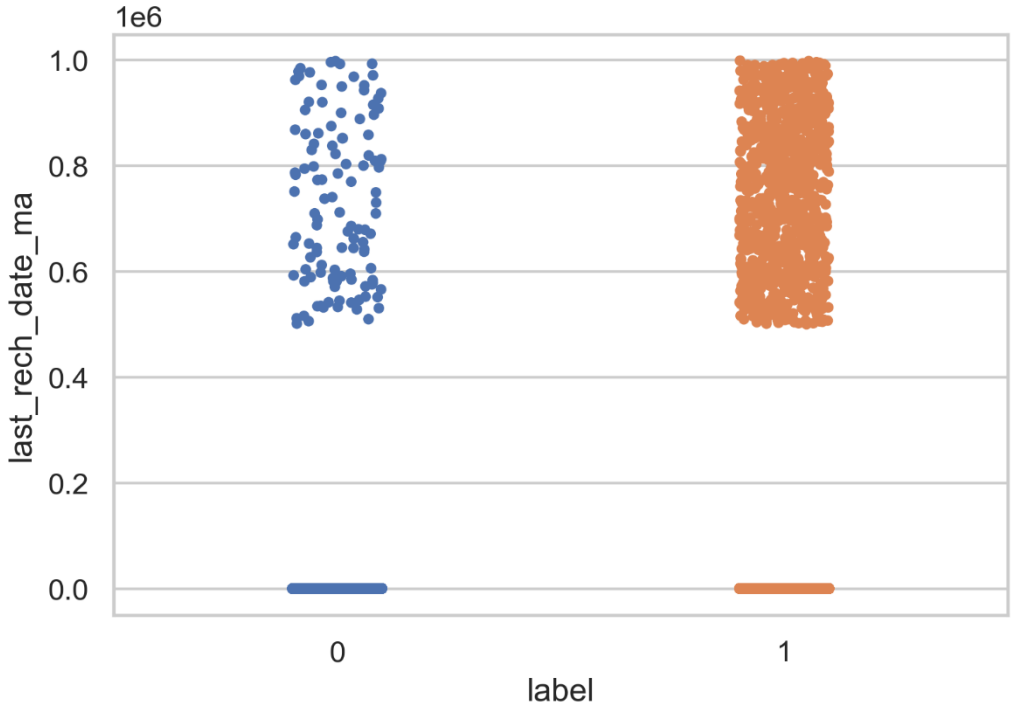


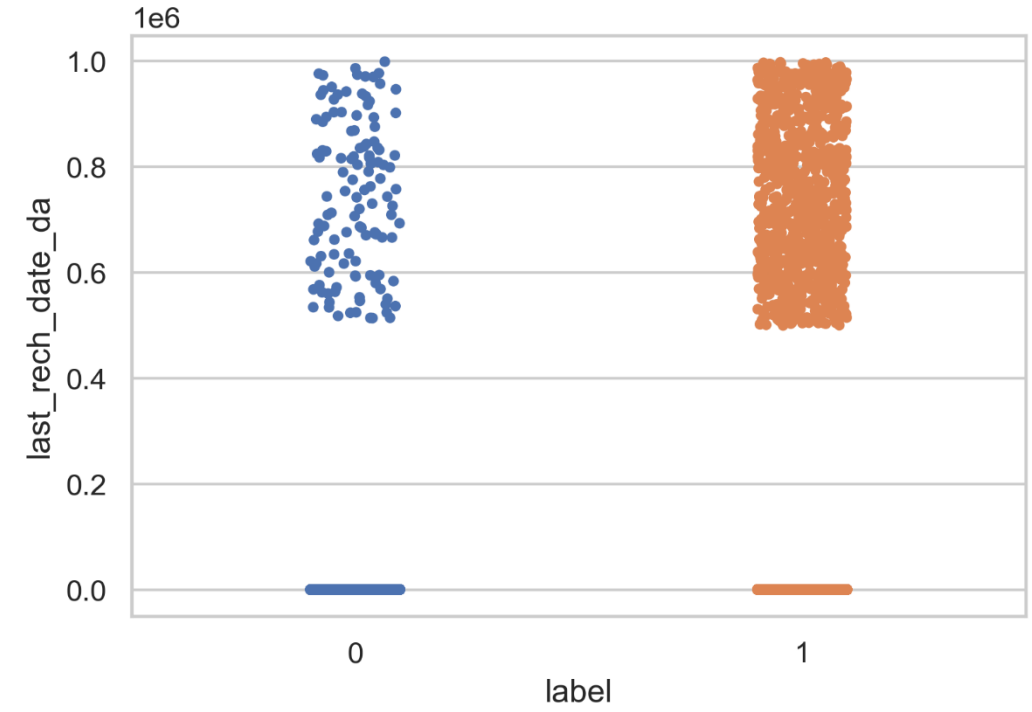


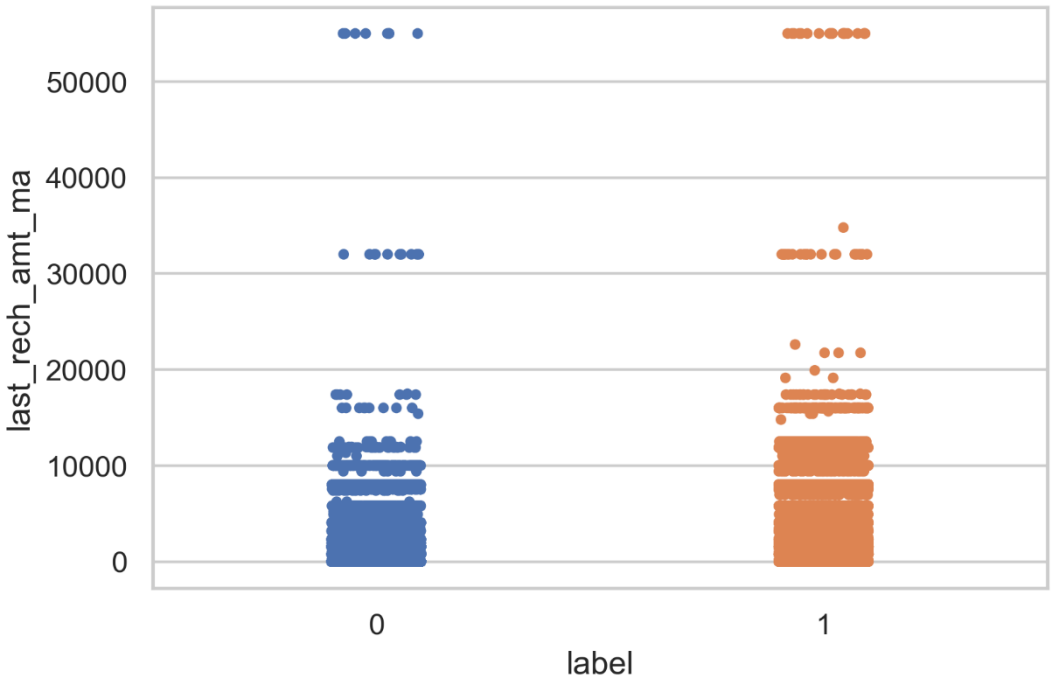

300000
250000
200000
150000
100000
50000
0
daily_decr90
0
1
label

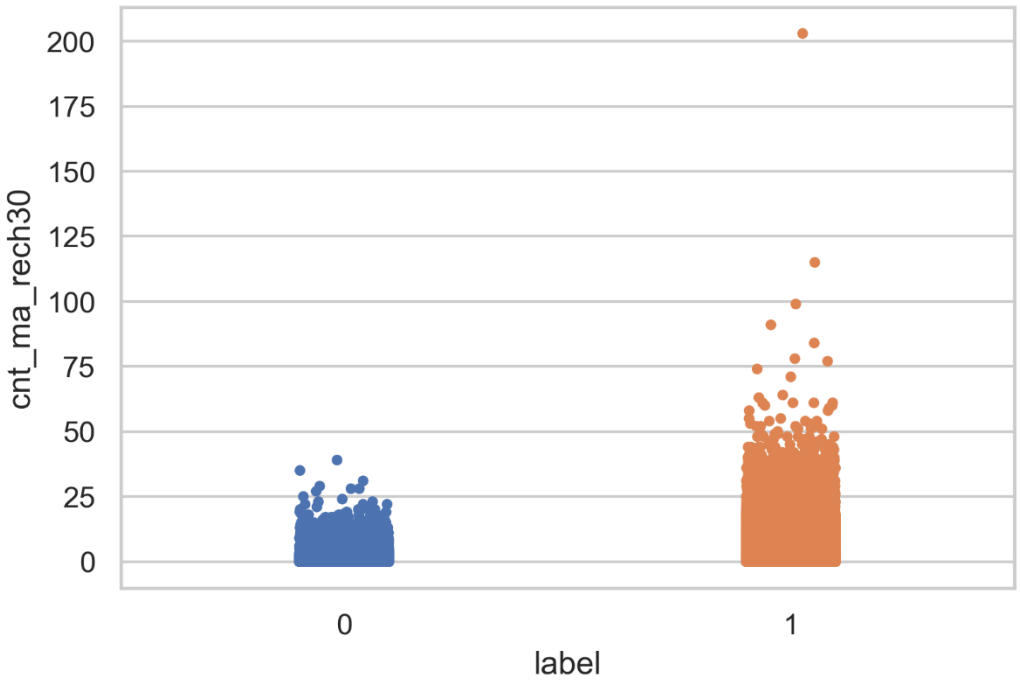

200000
175000
150000
125000
100000
75000
50000
25000
0
rental30
0
1
label

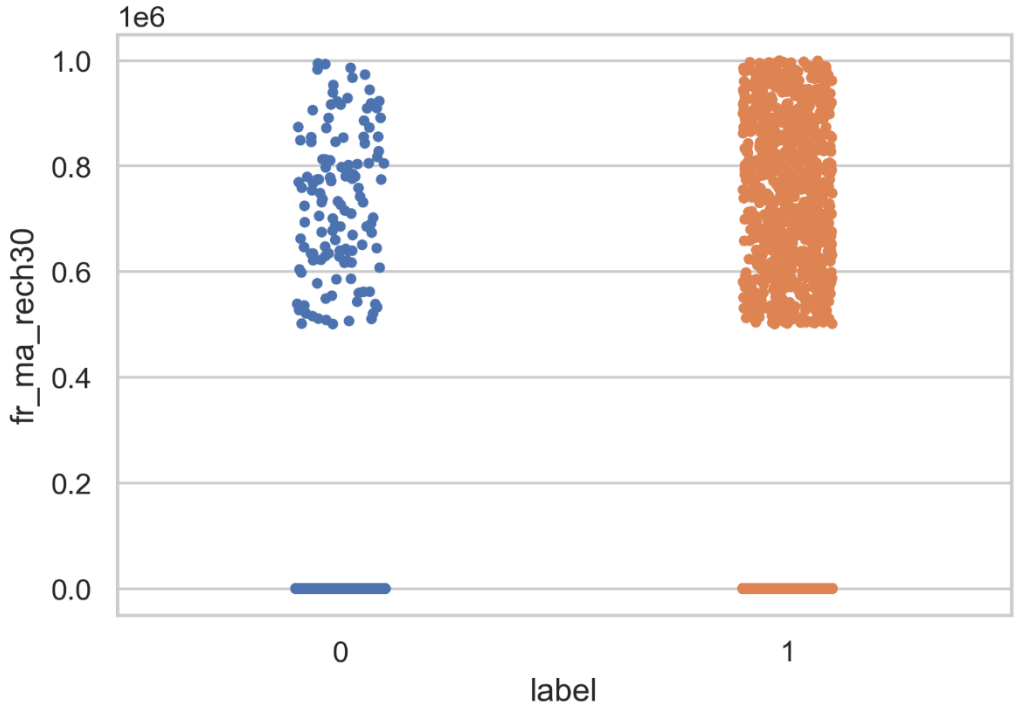

200000
175000
150000
125000
100000
75000
50000
25000
0
rental90
0
1
label

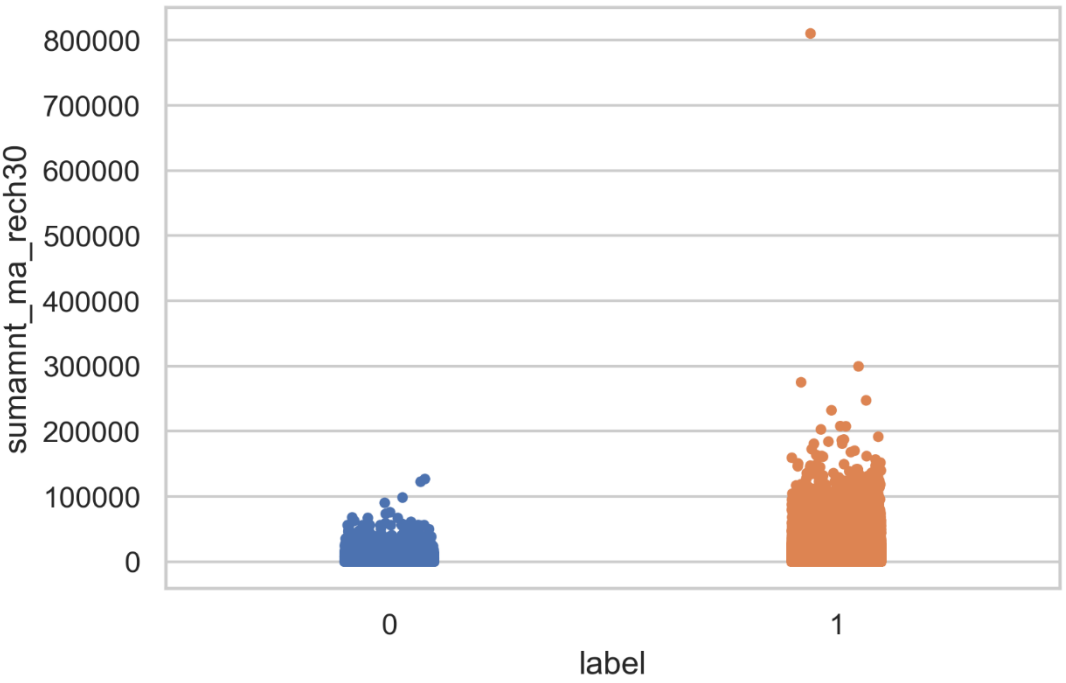

1e6
1.0
0.8
0.6
0.4
0.2
0.0
last_rech_date_ma
0
1
label

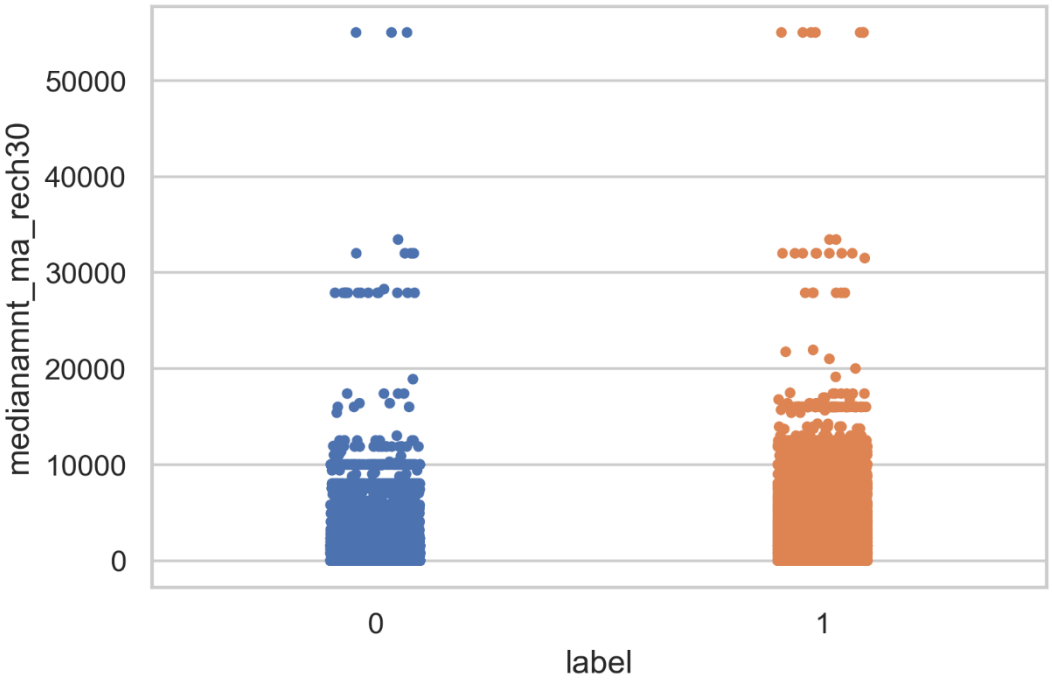

1e6
1.0
0.8
0.6
0.4
0.2
0.0
last_rech_date_da
0
1
label

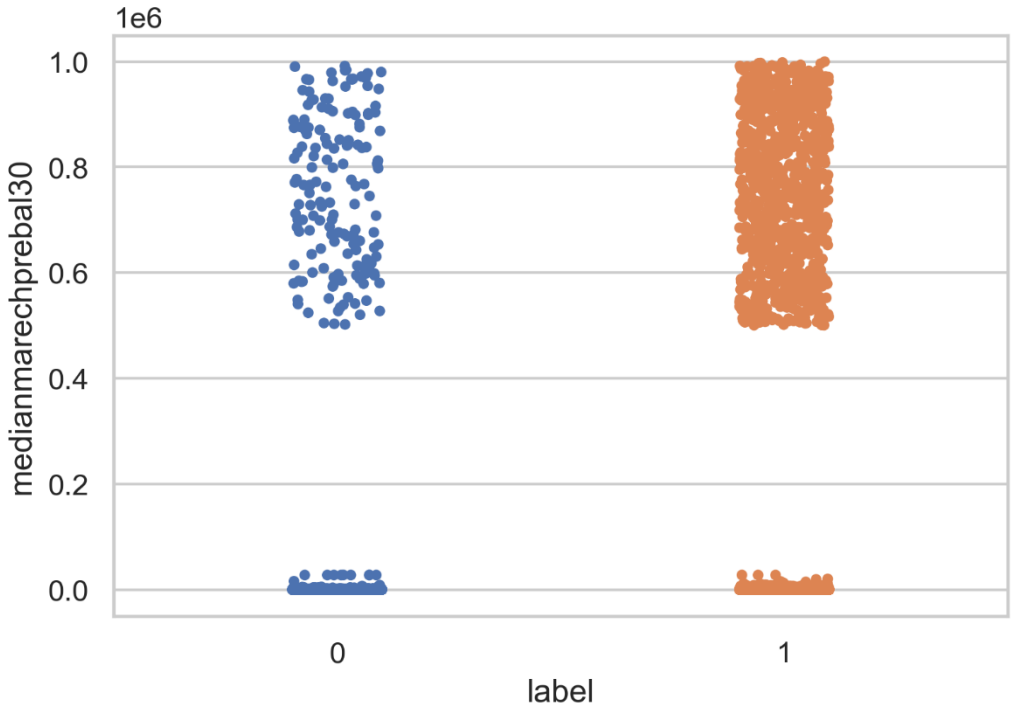

50000
40000
30000
20000
10000
0
last_rech_amt_ma
0
1
label

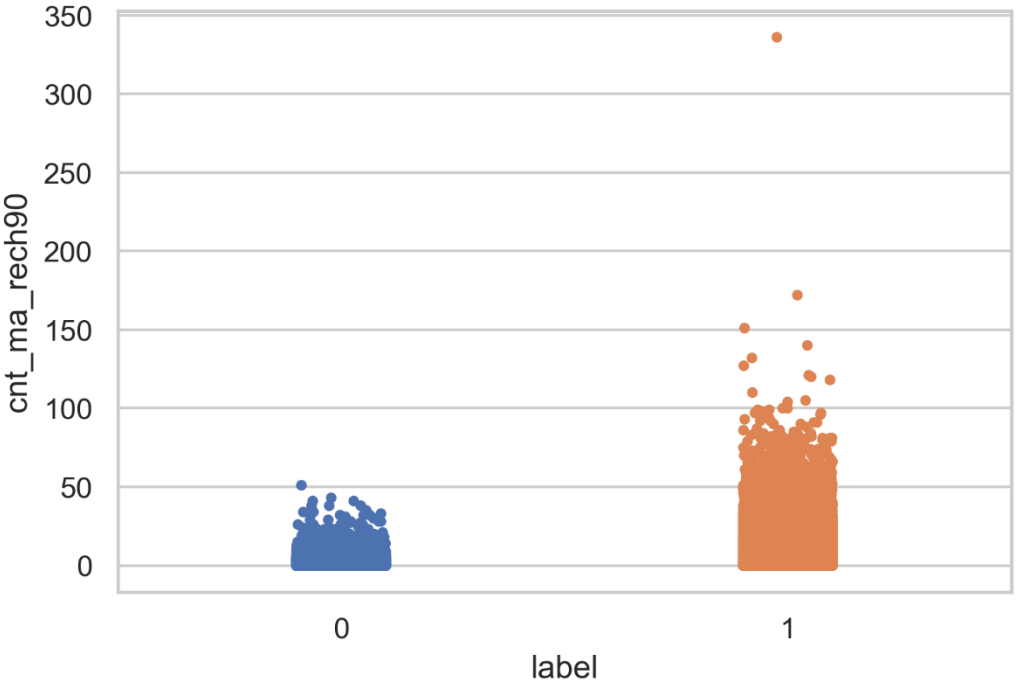

200
175
150
125
100
75
50
25
0
cnt_ma_rech30
0
1
label

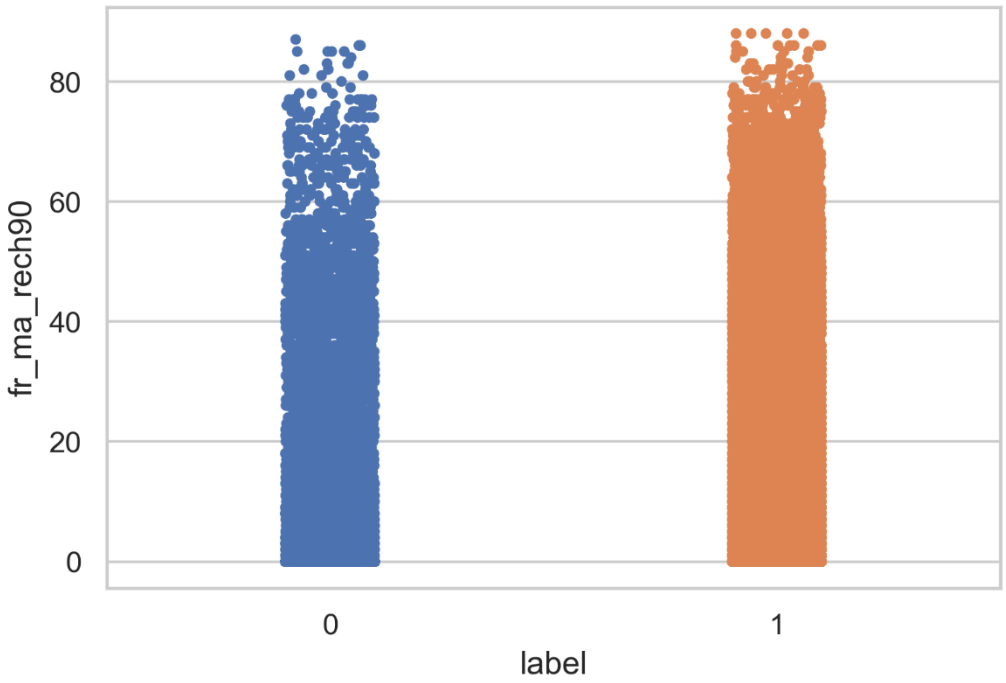

1e6
1.0
0.8
0.6
0.4
0.2
0.0
fr_ma_rech30
0
1
label

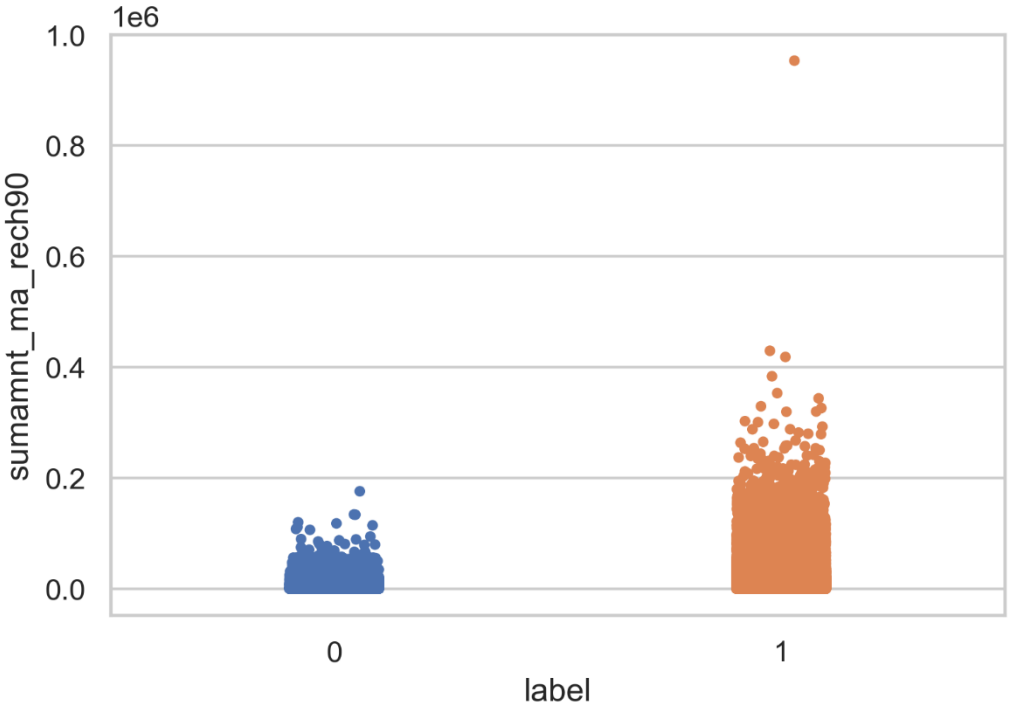

800000
700000
600000
500000
400000
300000
200000
100000
0
sumamnt_ma_rech30
0
1
label

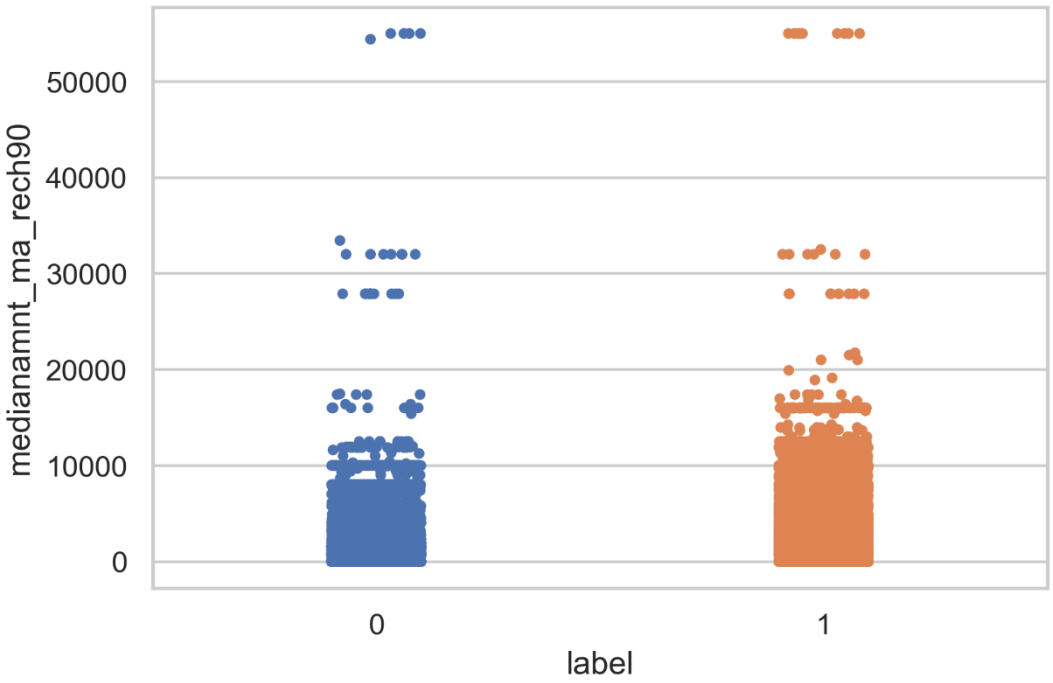

50000
40000
30000
20000
10000
0
medianamnt_ma_rech30
0
1
label

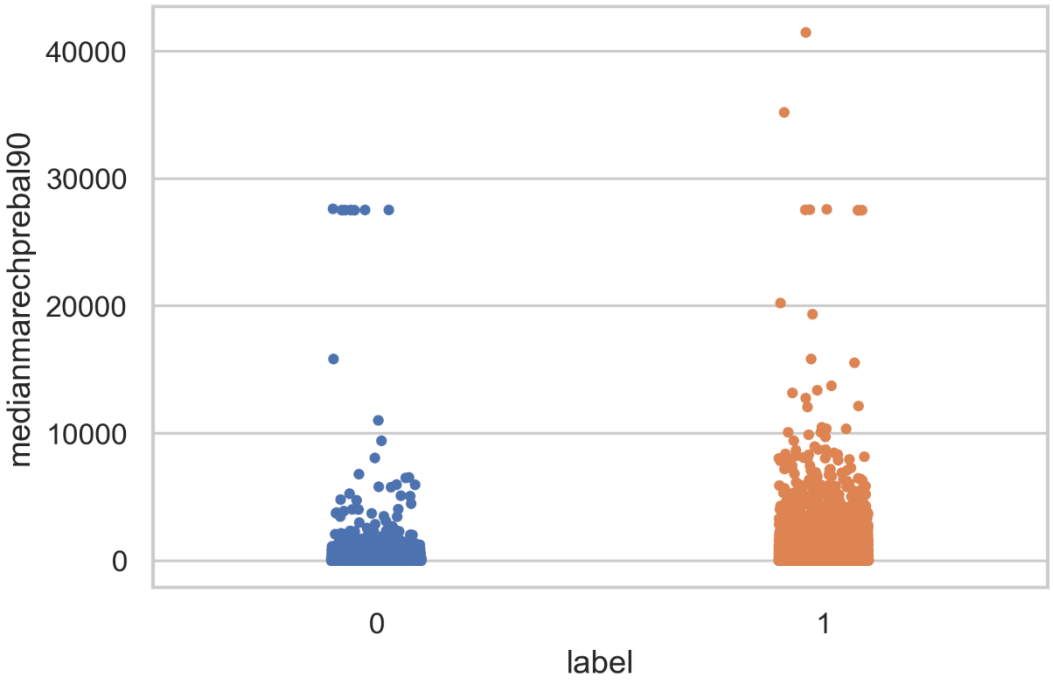

1e6
medianmarechpreba130
1.0
0.8
0.6
0.4
0.2
0.0
0
1
label

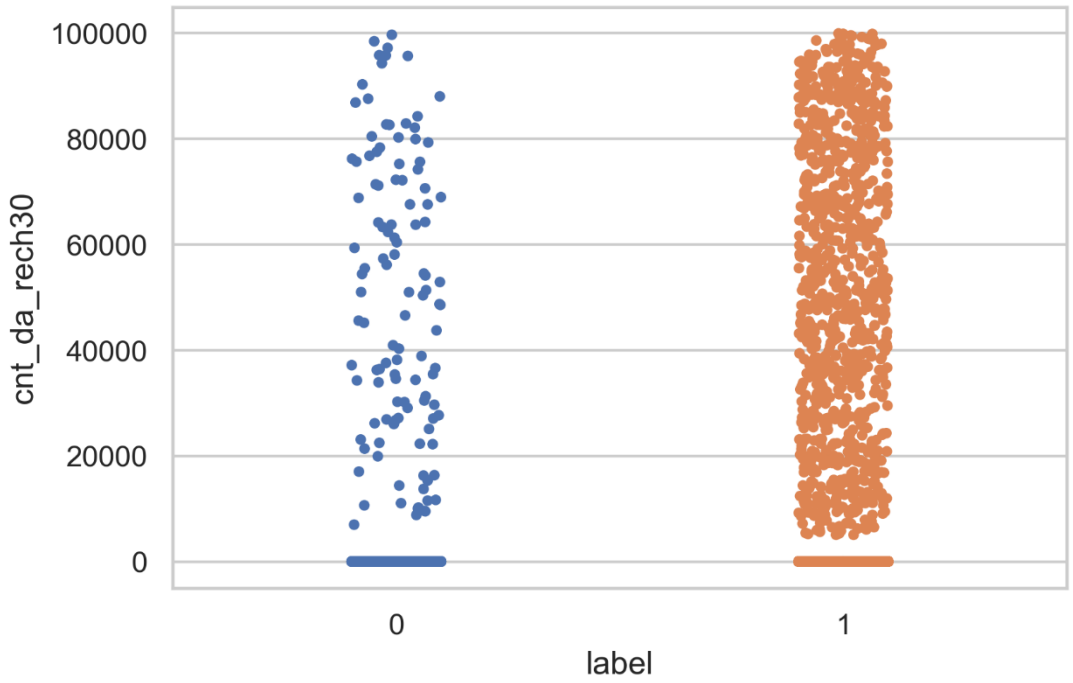

cnt_ma_rech90
350
300
250
200
150
100
50
0
0
1
label

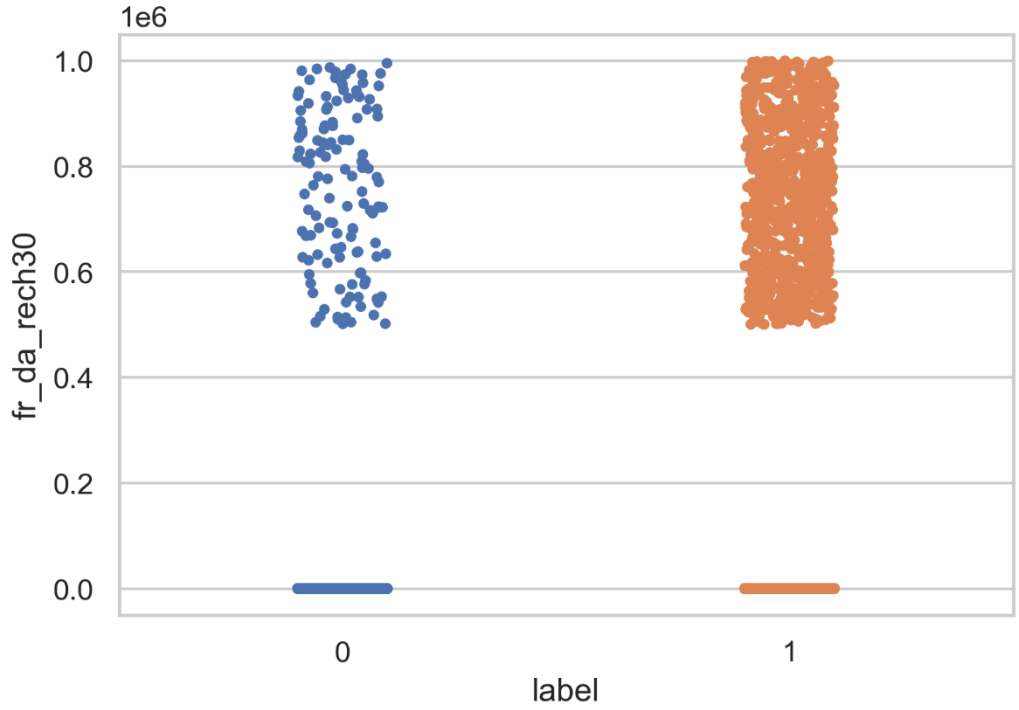

fr_ma_rech90
80
60
40
20
0
0
1
label

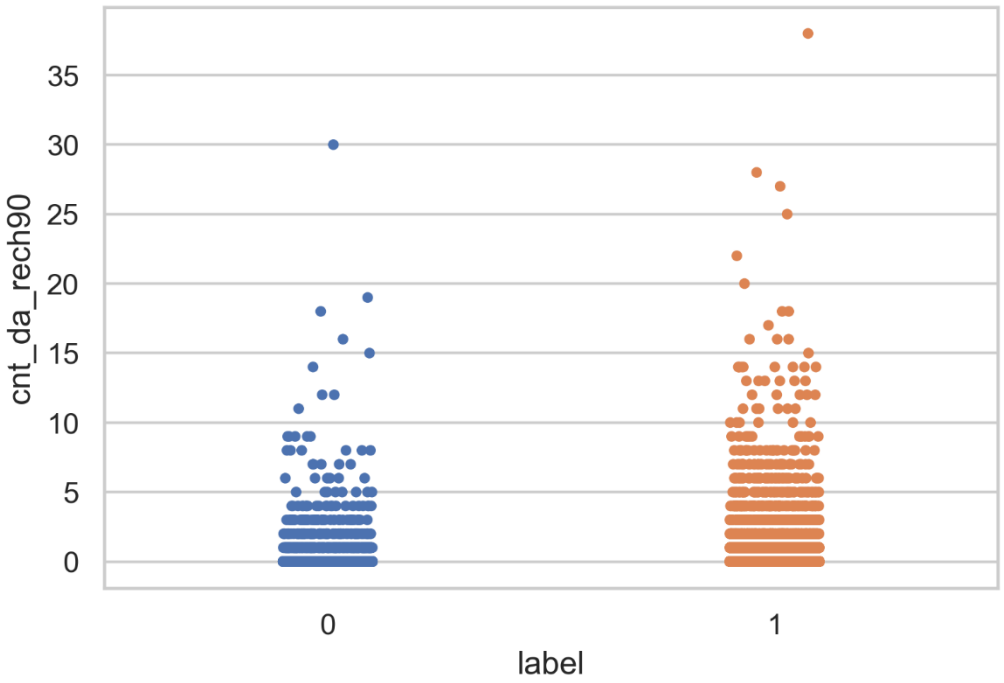

1e6
1.0
0.8
0.6
0.4
0.2
0.0
sumamnt_ma_rech90
0
1
label

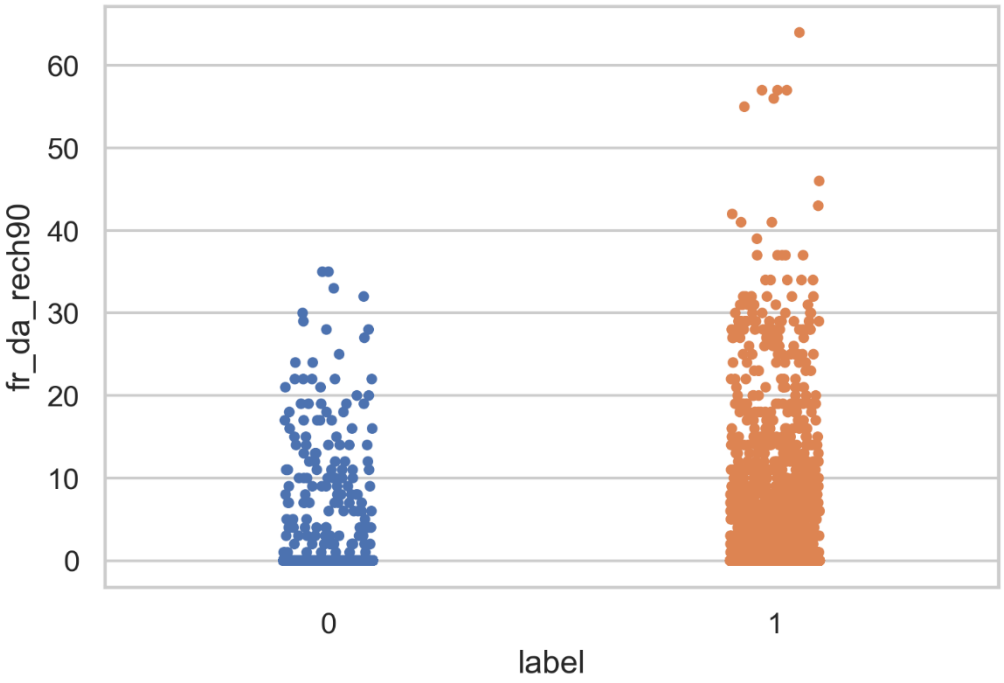

medianamnt_ma_rech90
50000
40000
30000
20000
10000
0
0
1
label

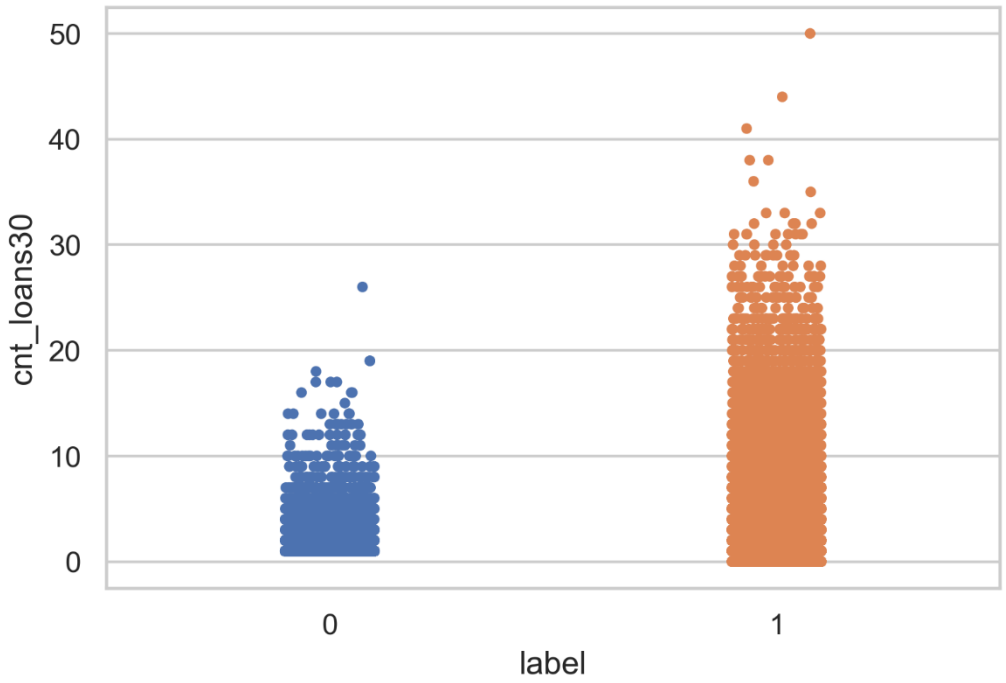

medianmarechpreba190
40000
30000
20000
10000
0
0
1
label

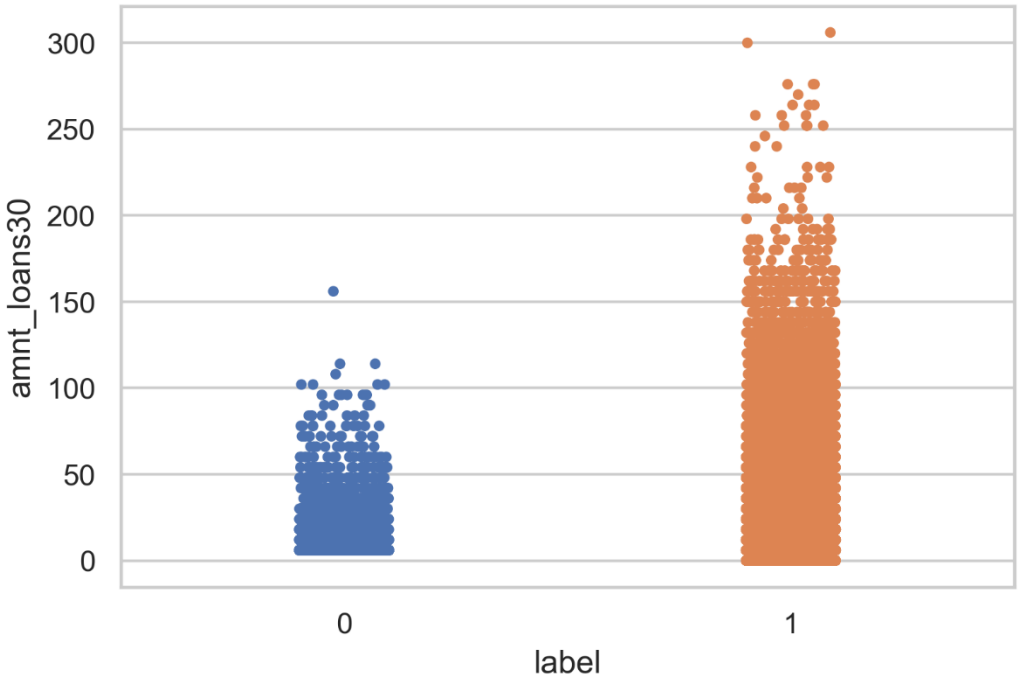

100000
80000
60000
40000
20000
0
cnt_da_rech30
0
1
label

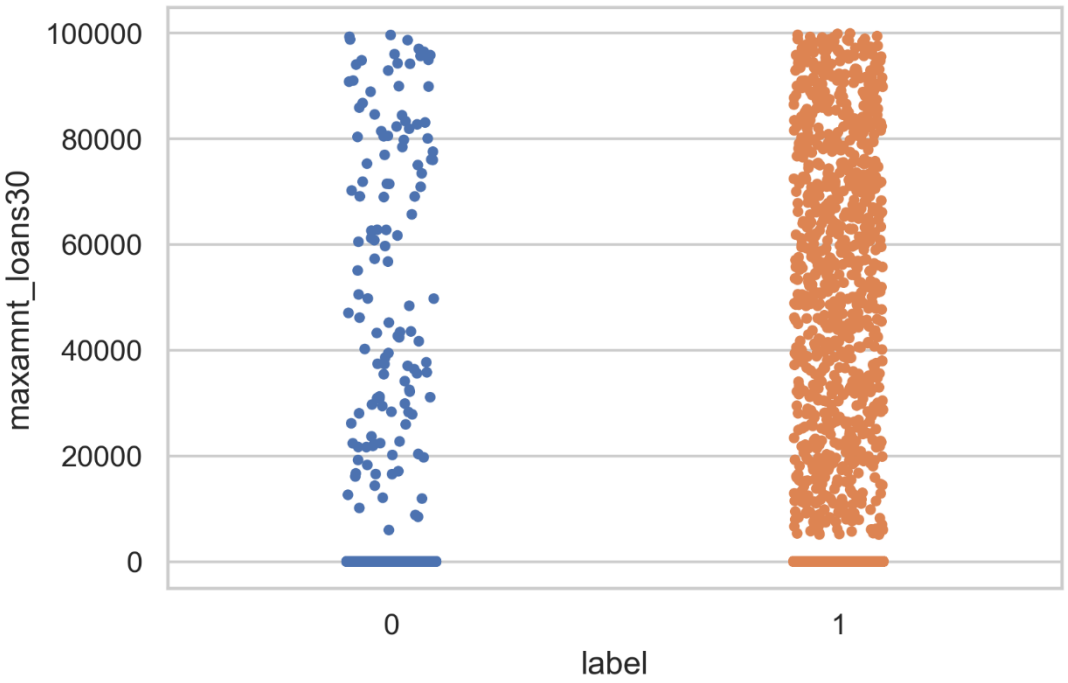

1e6
1.0
0.8
0.6
0.4
0.2
0.0
fr_da_rech30
0
1
label

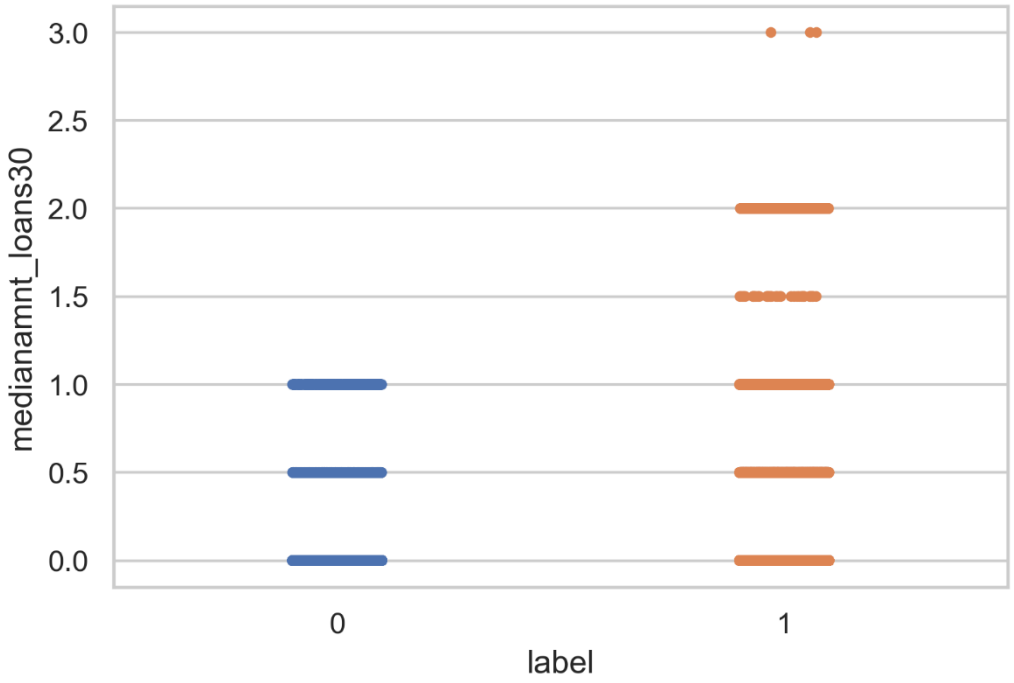

cnt_da_rech90
35
30
25
20
15
10
5
0
0
1
label

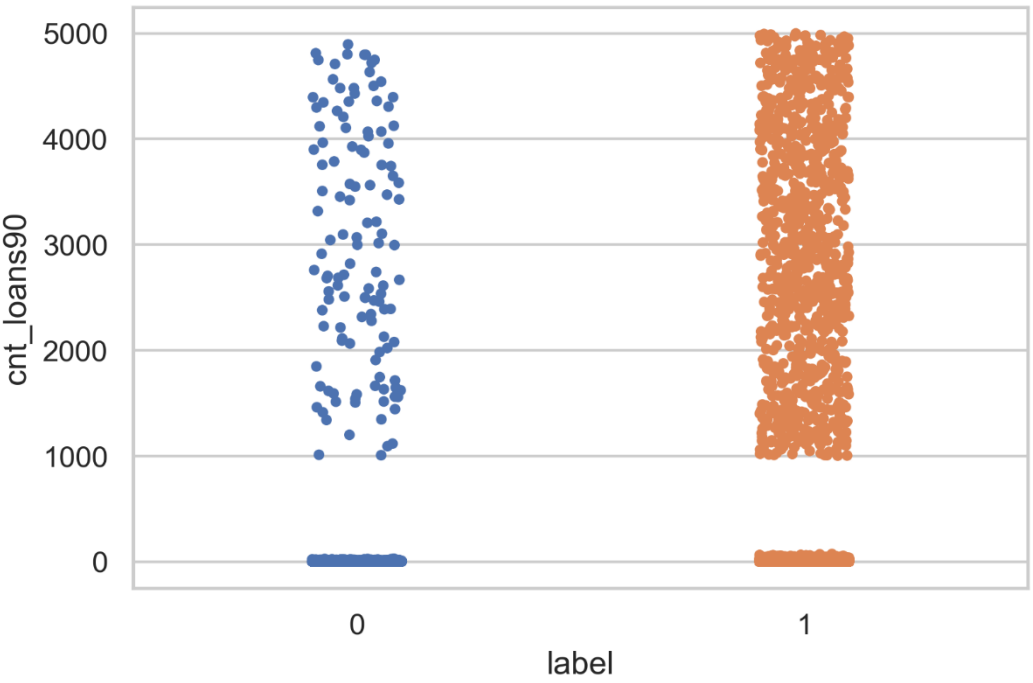

fr_da_rech90
60
50
40
30
20
10
0
0
1
label

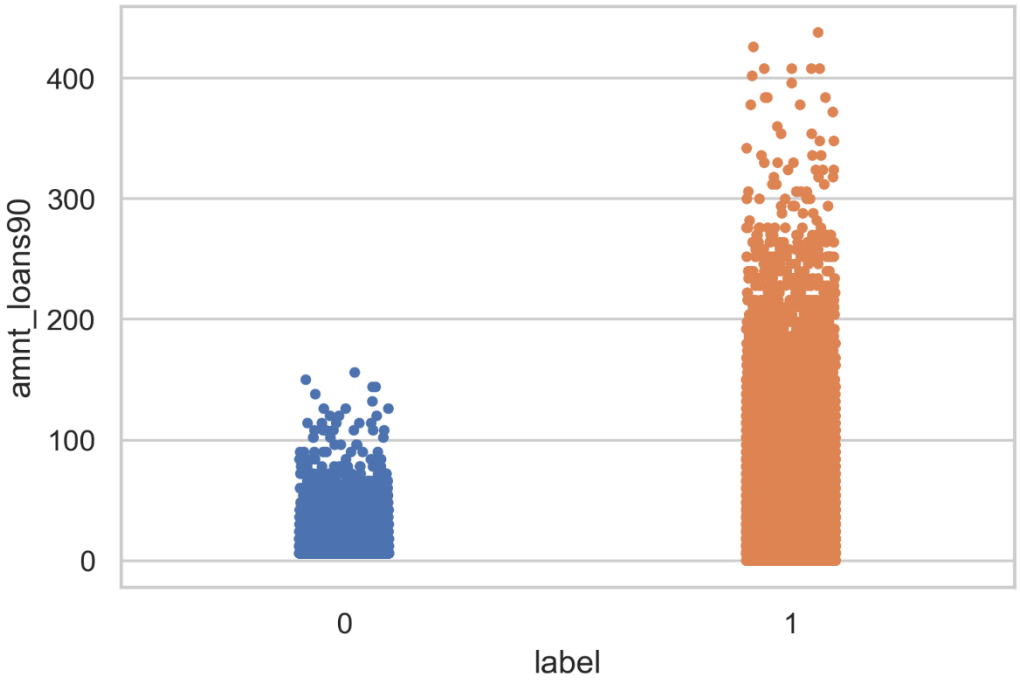

50
40
30
20
10
0
cnt_loans30
0
1
label

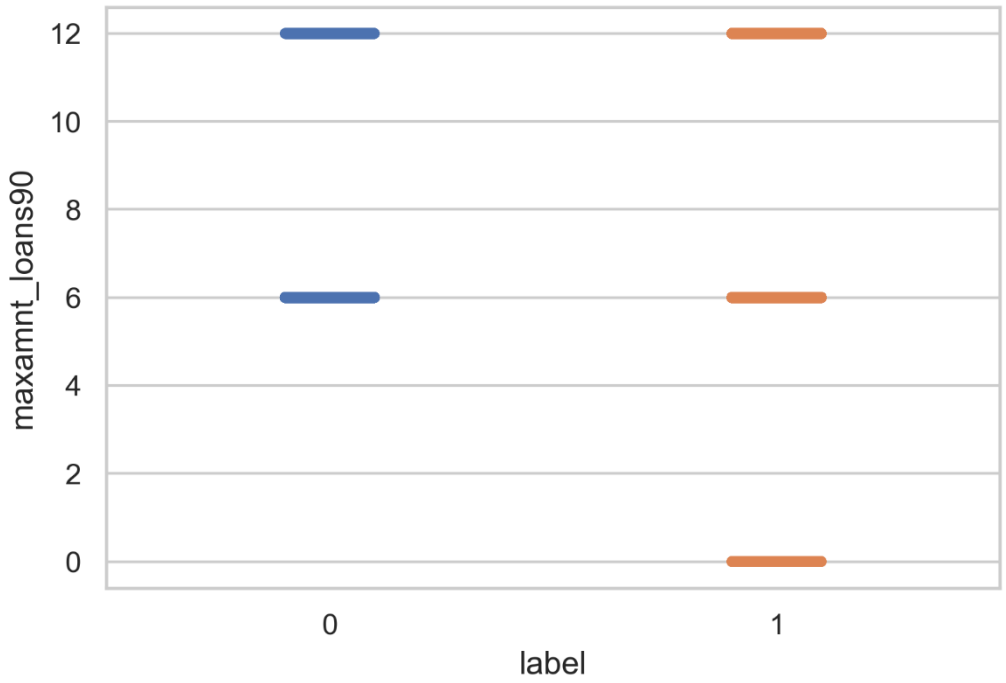

300
250
200
150
100
50
0
amnt_loans30
0
1
label

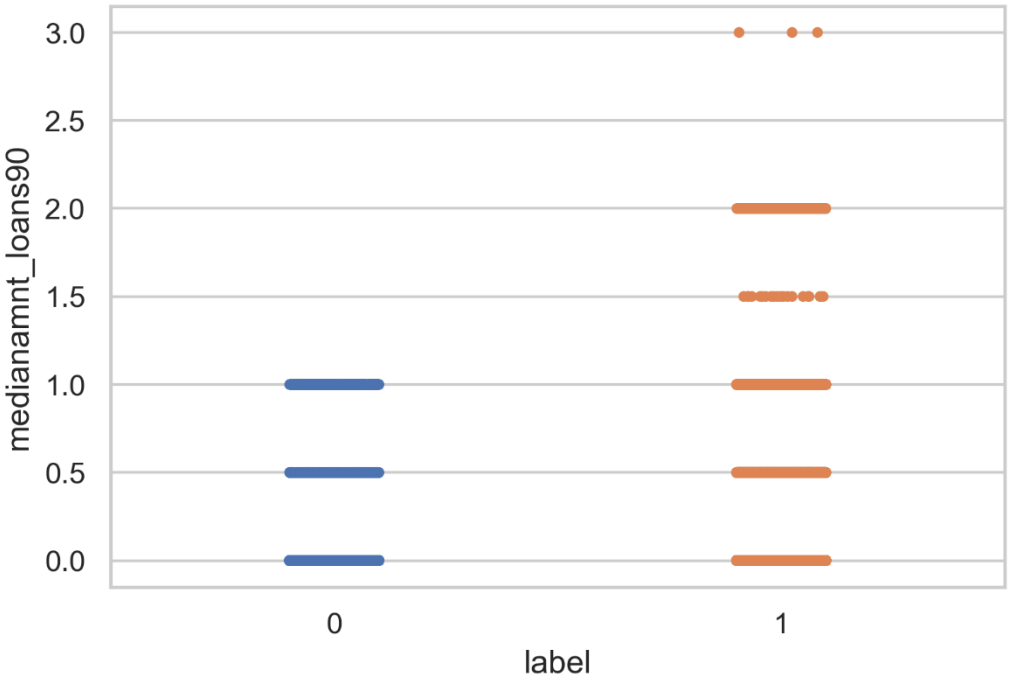

100000
80000
60000
40000
20000
0
maxamnt_loans30
0
1
label

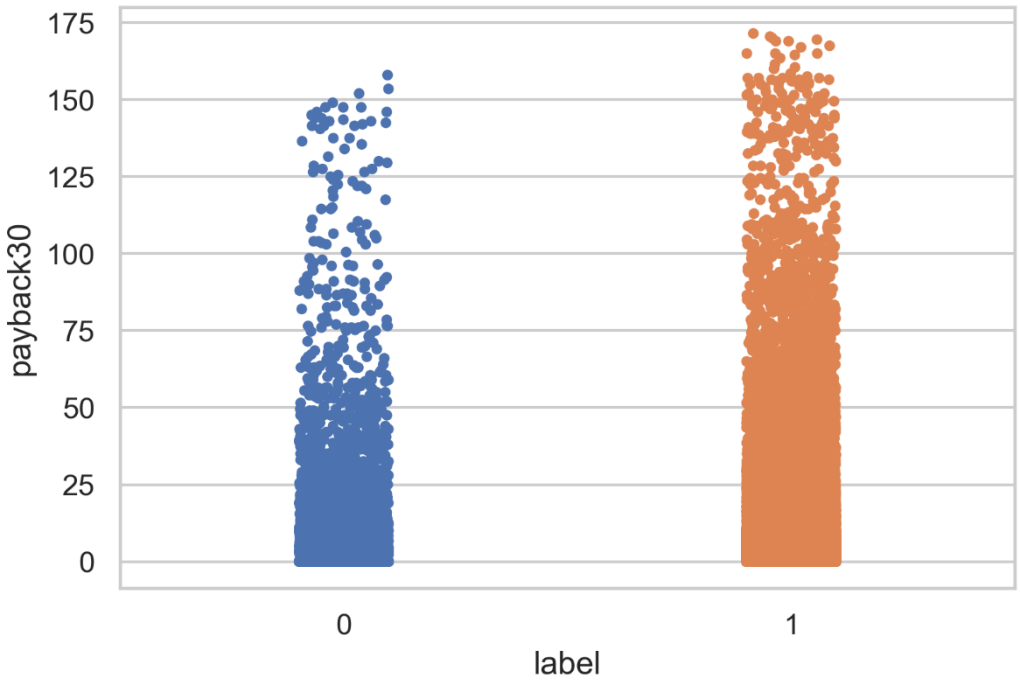

3.0
2.5
2.0
1.5
1.0
0.5
0.0
medianamnt_loans30
0
1
label

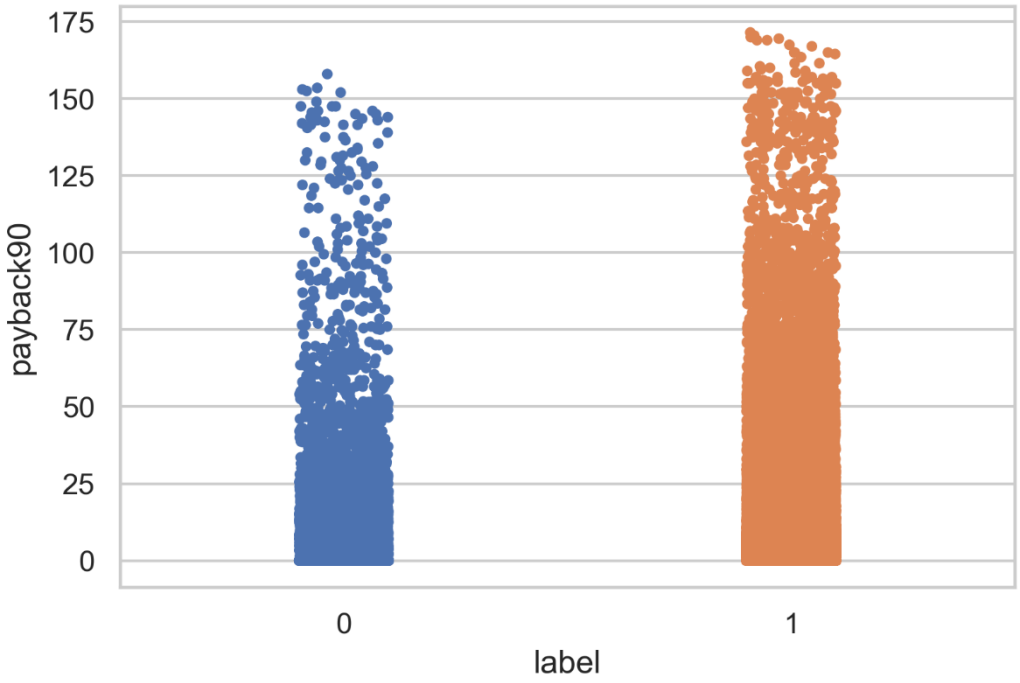

5000
4000
3000
2000
1000
0
cnt_loans90
0
1
label

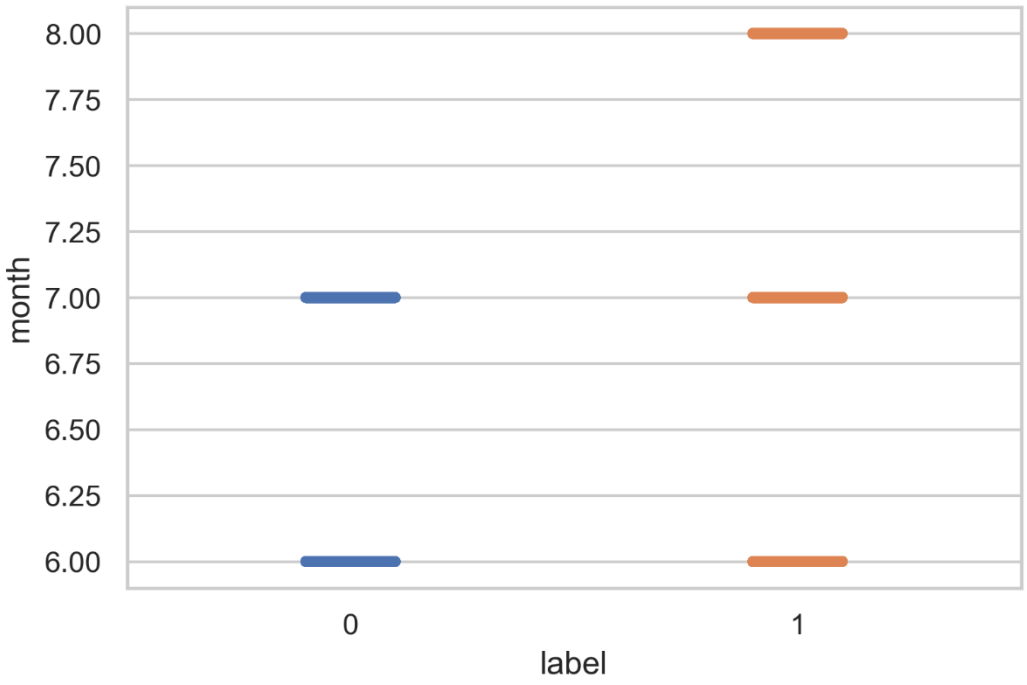

400
300
200
100
0
amnt_loans90
0
1
label

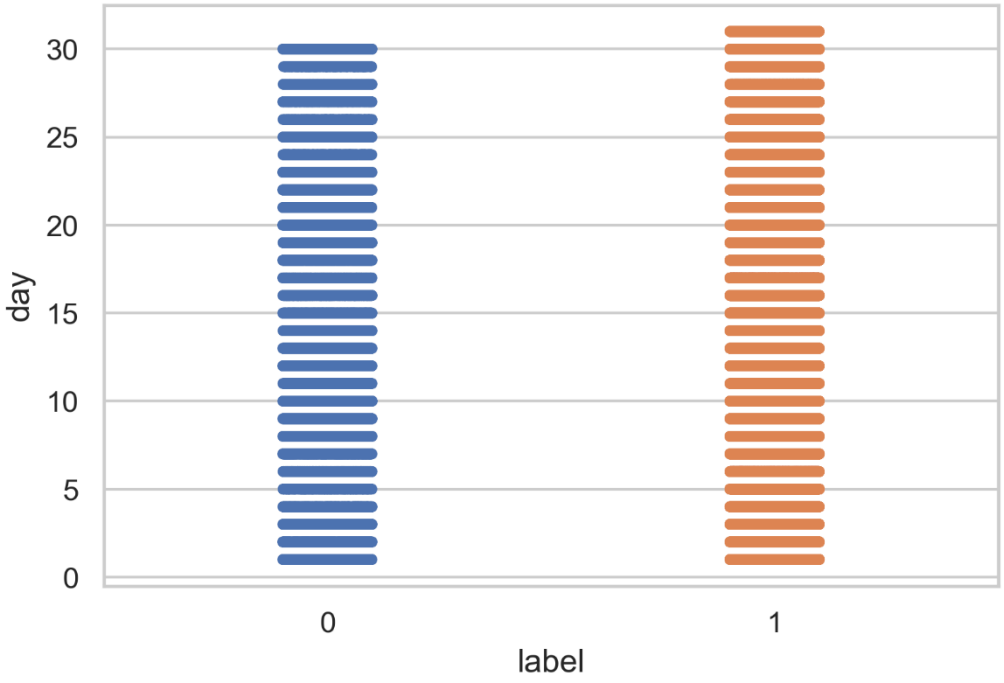

maxamnt_loans90
12
10
8
6
4
2
0
0
1
label

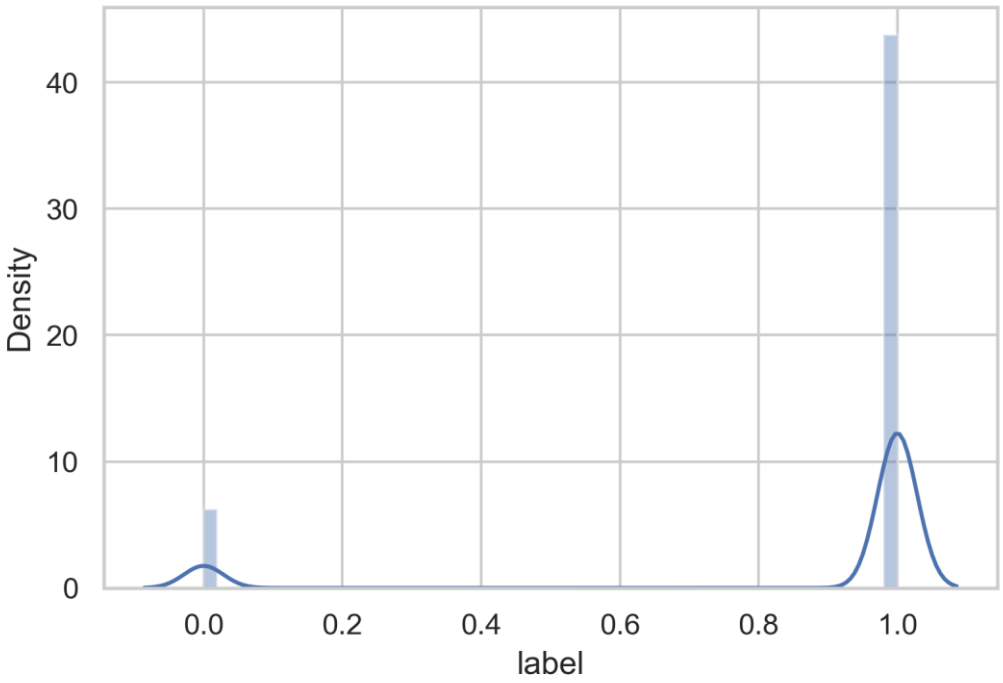

medianamnt_loans90
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
label

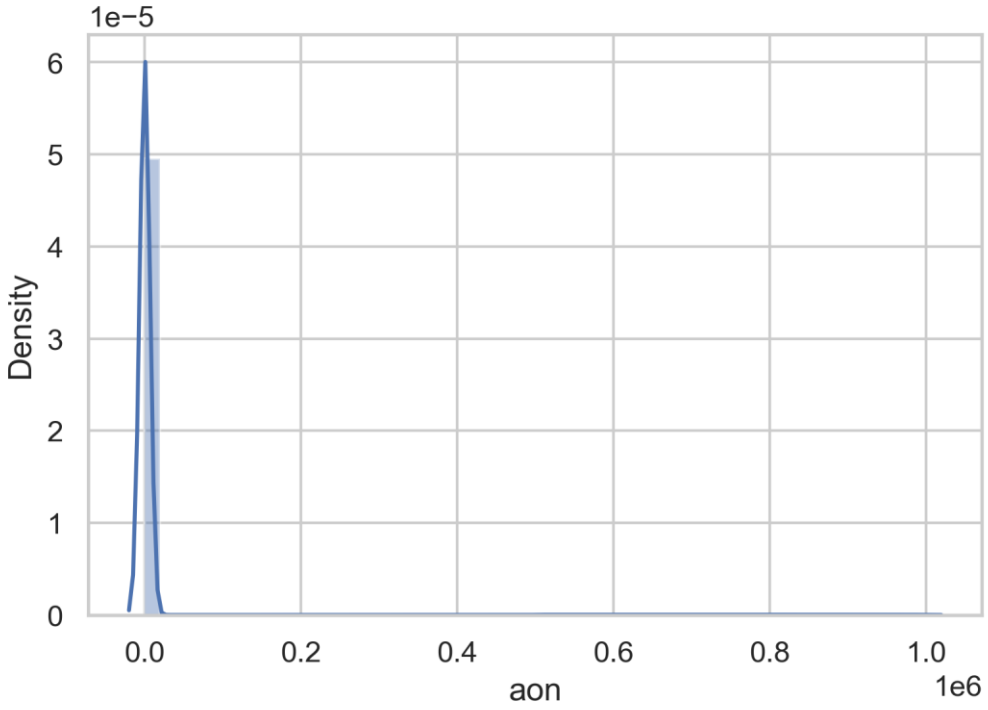

175
150
125
100
75
50
25
0
payback30
0
1
label

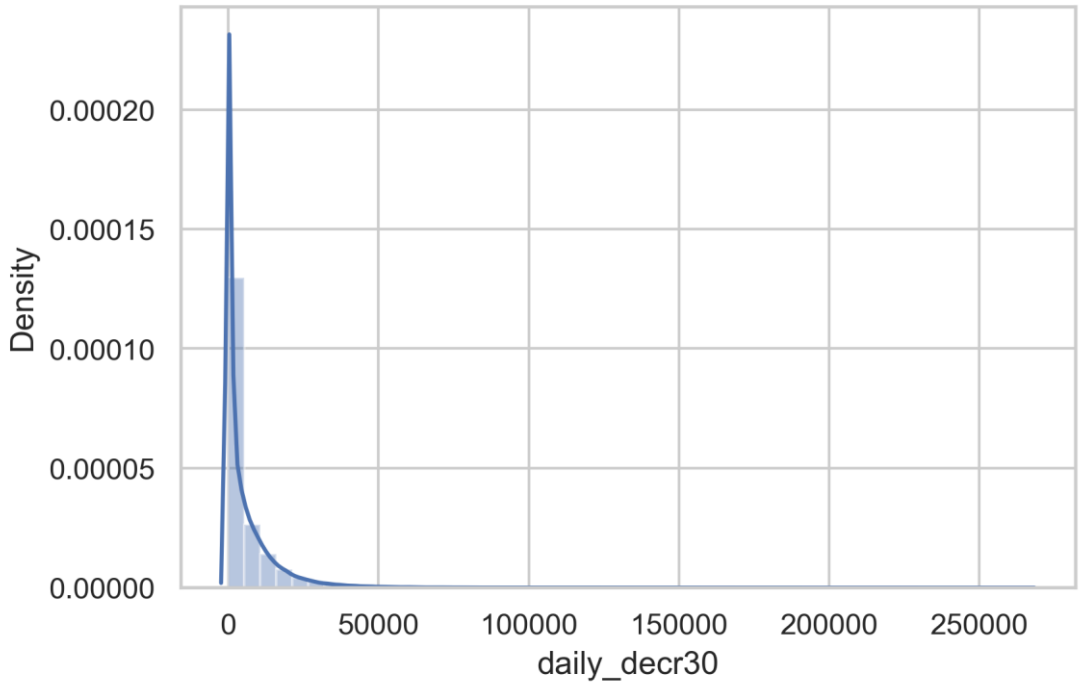

175
150
125
100
75
50
25
0
payback90
0
1
label

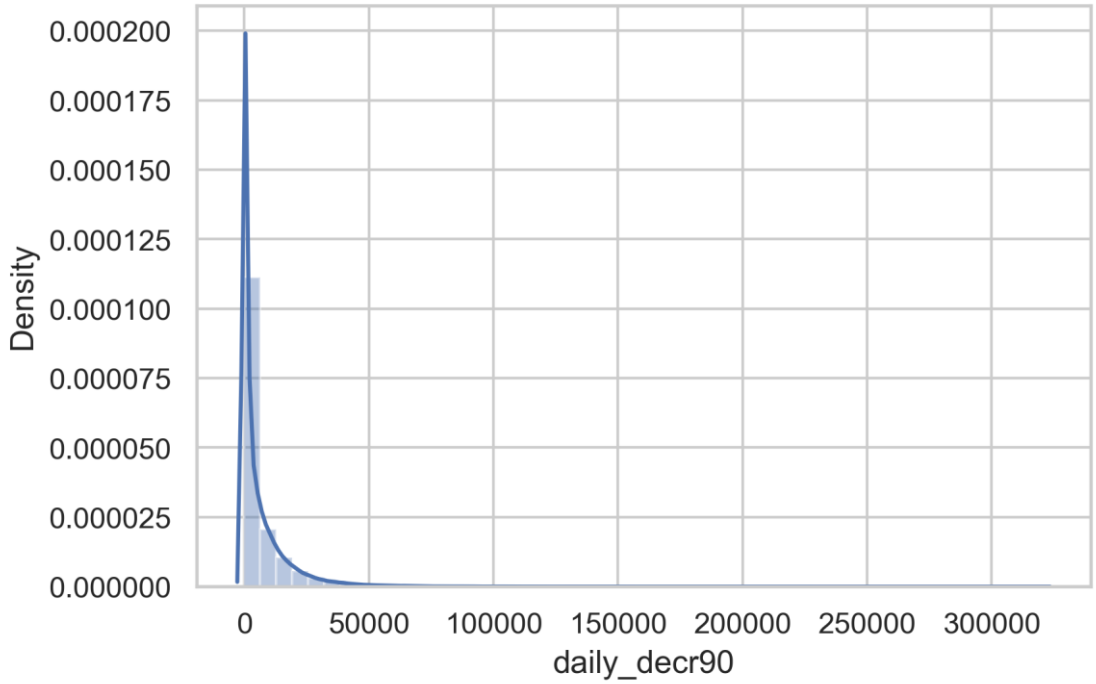

8.00
7.75
7.50
7.25
7.00
6.75
6.50
6.25
6.00
month
0
1
label

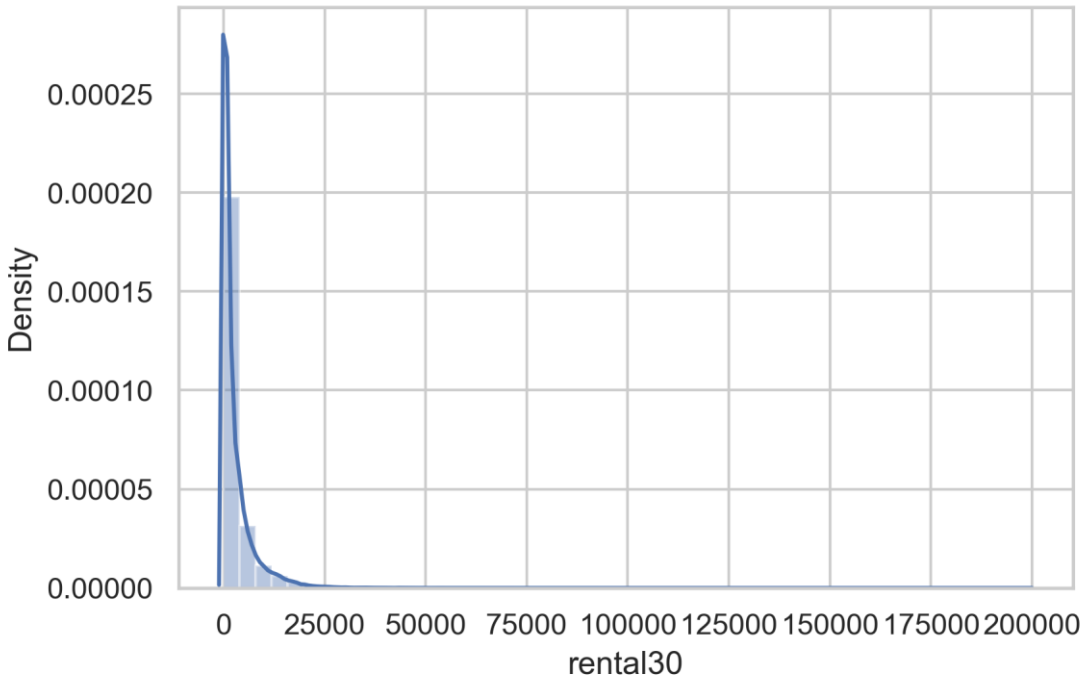

30
25
20
15
10
5
0
day
0
1
label

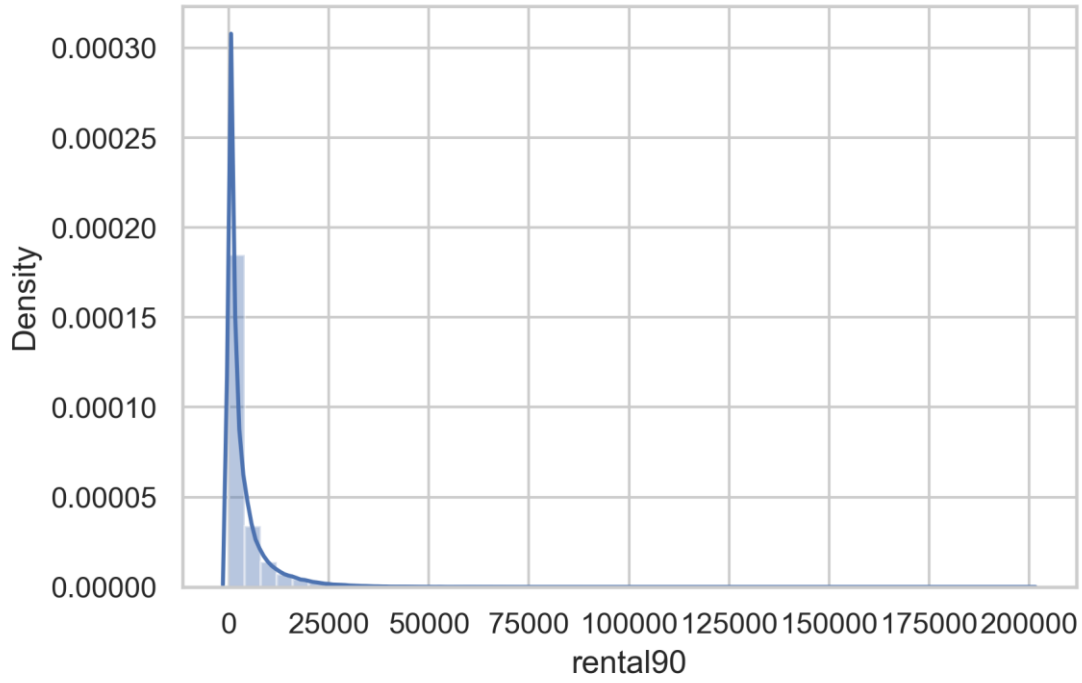

40
30
20
10
0
Density
0.0
0.2
0.4
0.6
0.8
1.0
label

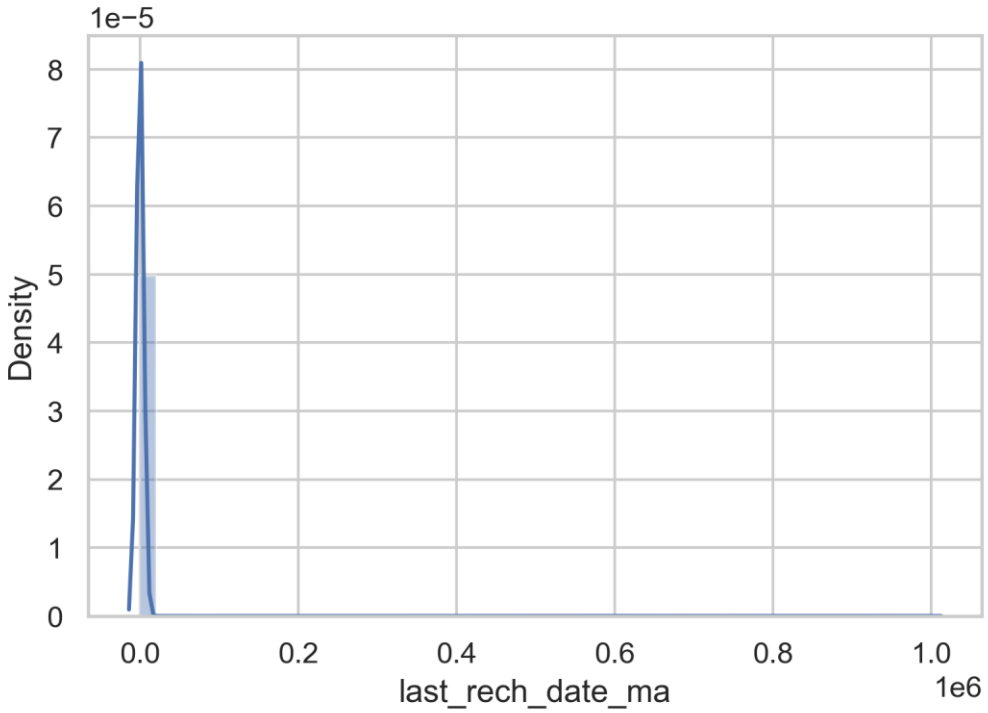

1e−5
6
5
4
3
2
1
0
Density
0.0
0.2
0.4
0.6
0.8
1.0
1e6
aon

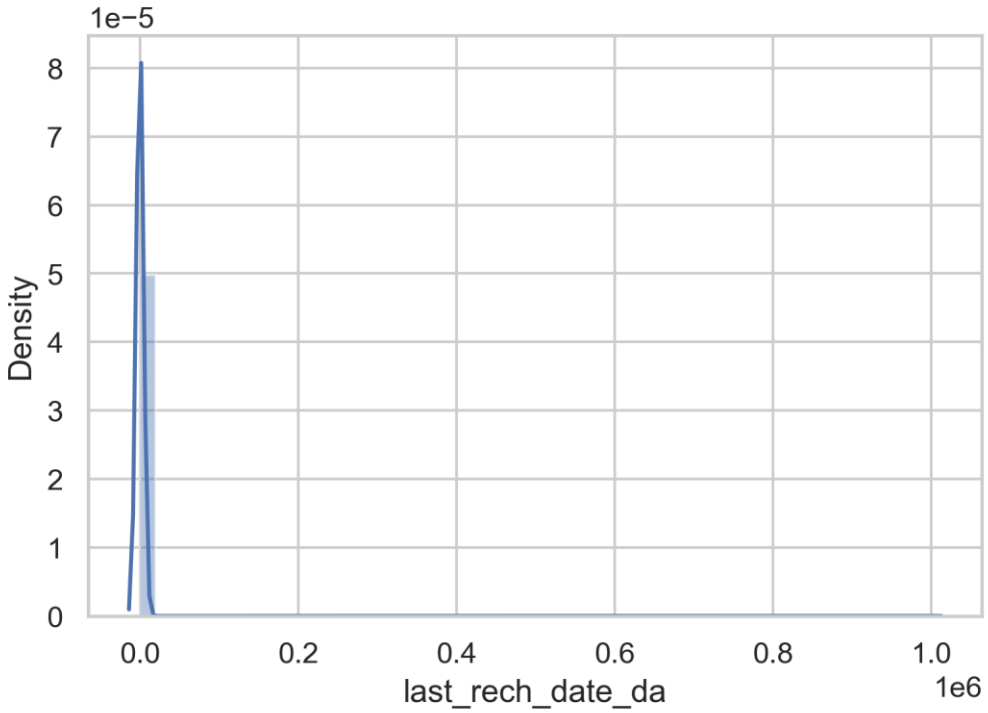

0.00020
0.00015
0.00010
0.00005
0.00000
Density
0
50000
100000
150000
200000
250000
daily_decr30

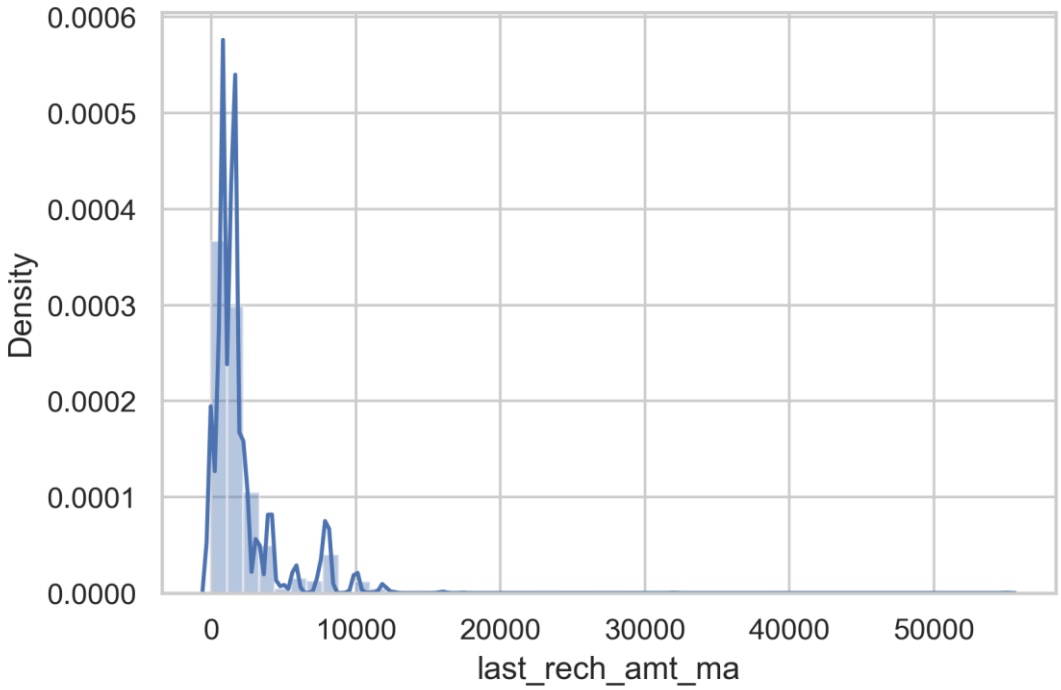

0.000200
0.000175
0.000150
0.000125
0.000100
0.000075
0.000050
0.000025
0.000000
Density
0
50000
100000
150000
200000
250000
300000
daily_decr90

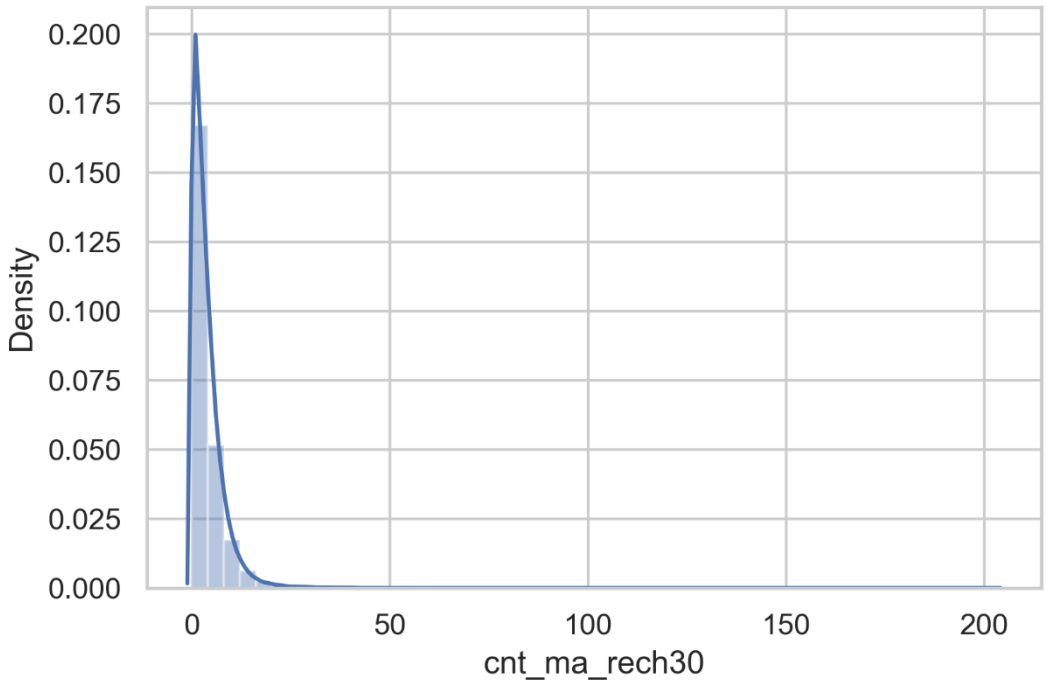

0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
Density
0
25000
50000
75000
100000
125000
150000
175000
200000
rental30

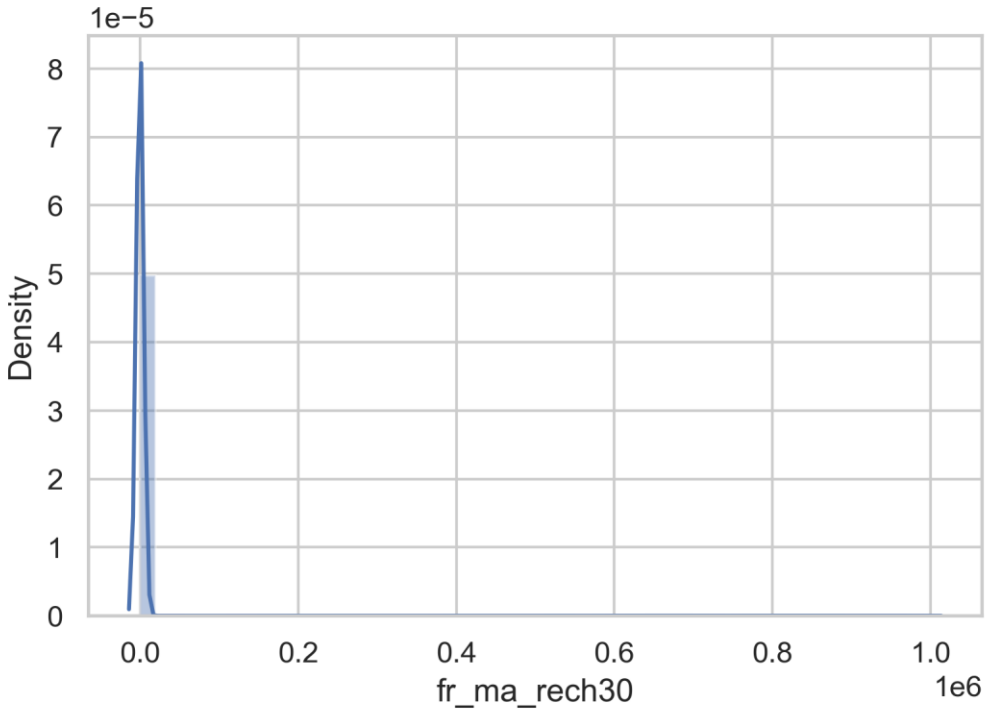

0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
Density
0
25000
50000
75000
100000
125000
150000
175000
200000
rental90

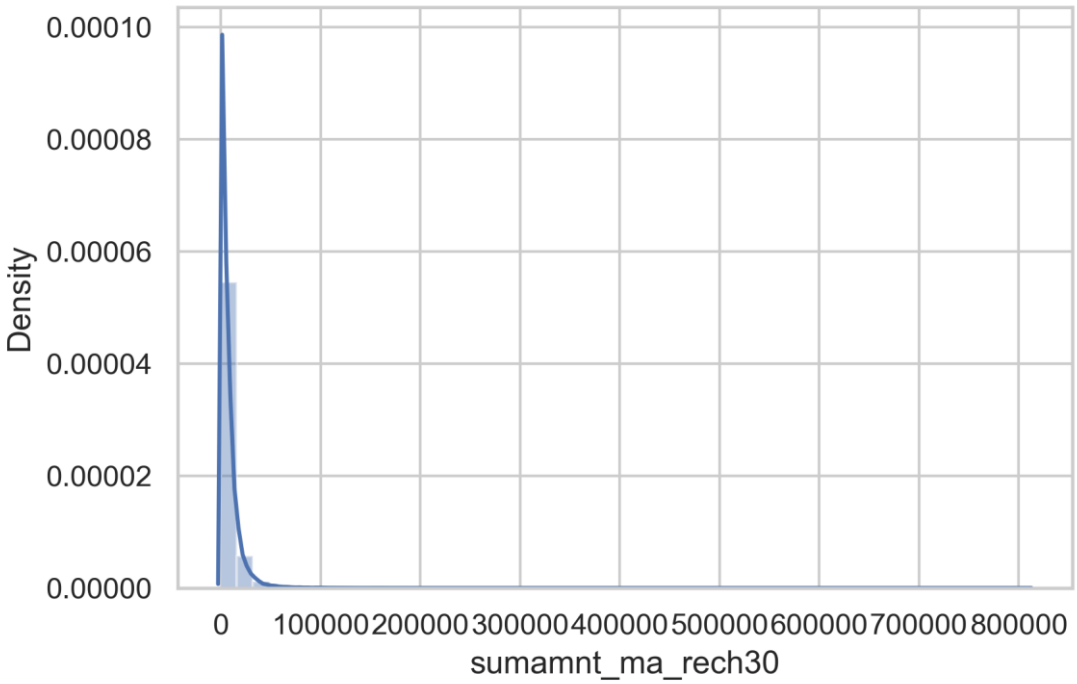

1e−5
8
7
6
5
4
3
2
1
0
Density
0.0
0.2
0.4
0.6
0.8
1.0
1e6
last_rech_date_ma

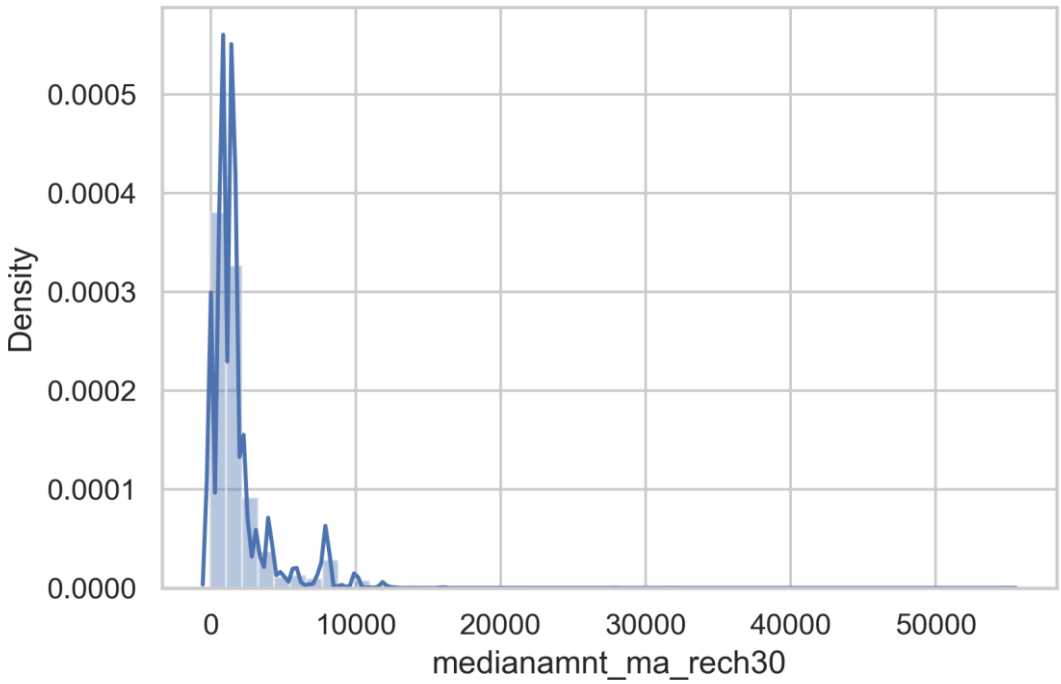

1e−5
8
7
6
5
4
3
2
1
0
Density
0.0
0.2
0.4
0.6
0.8
1.0
1e6
last_rech_date_da

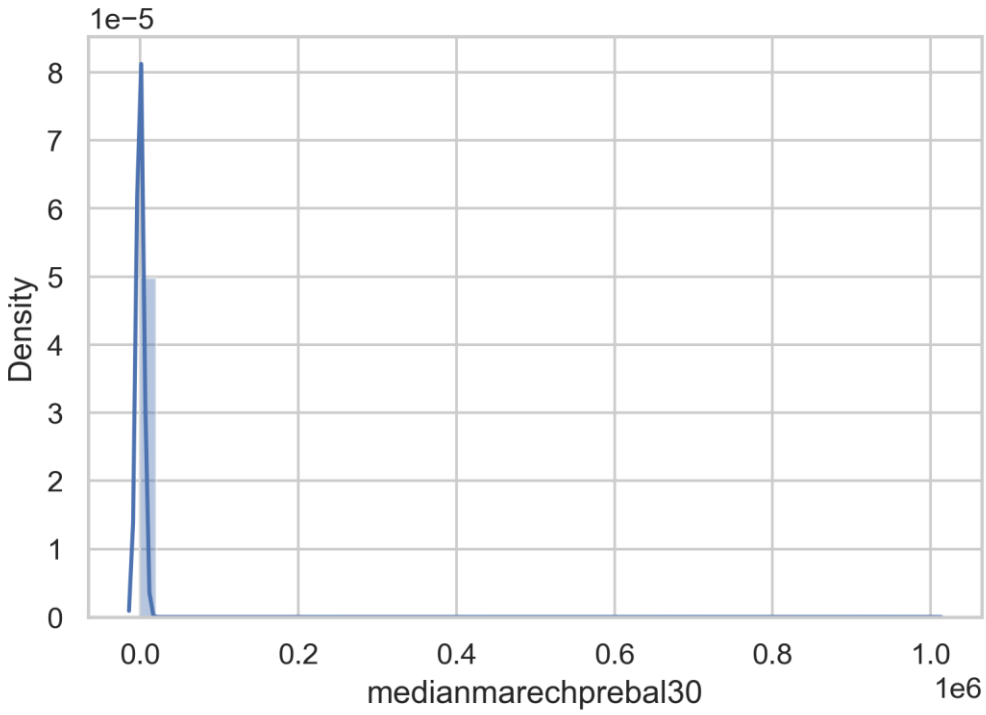

0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
Density
0
10000
20000
30000
40000
50000
last_rech_amt_ma

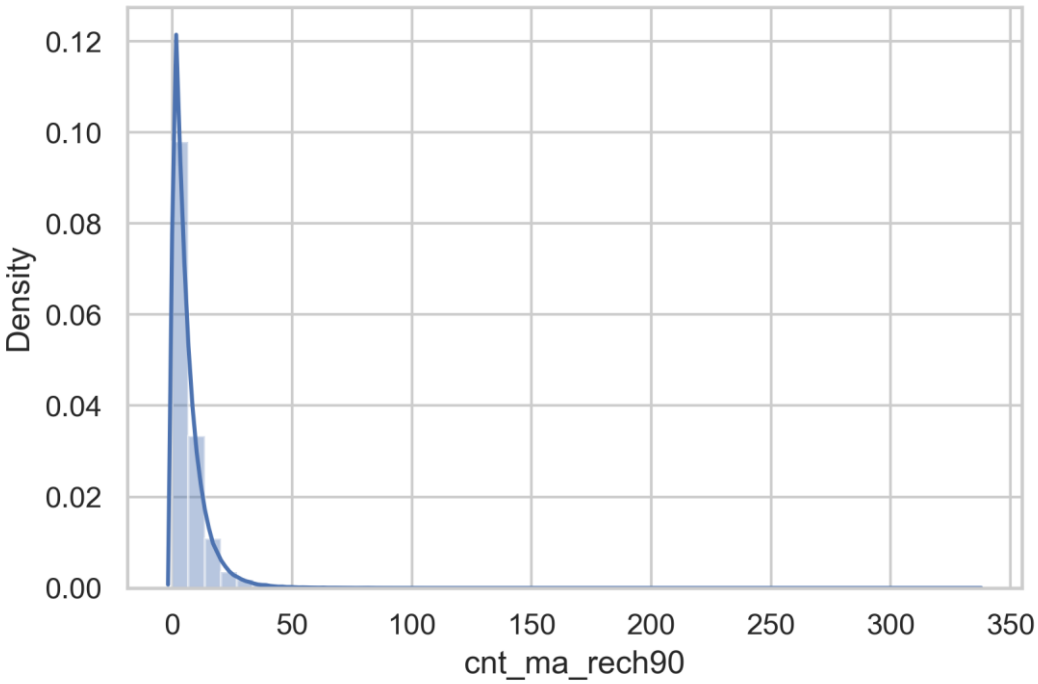

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Density
0
50
100
150
200
cnt_ma_rech30

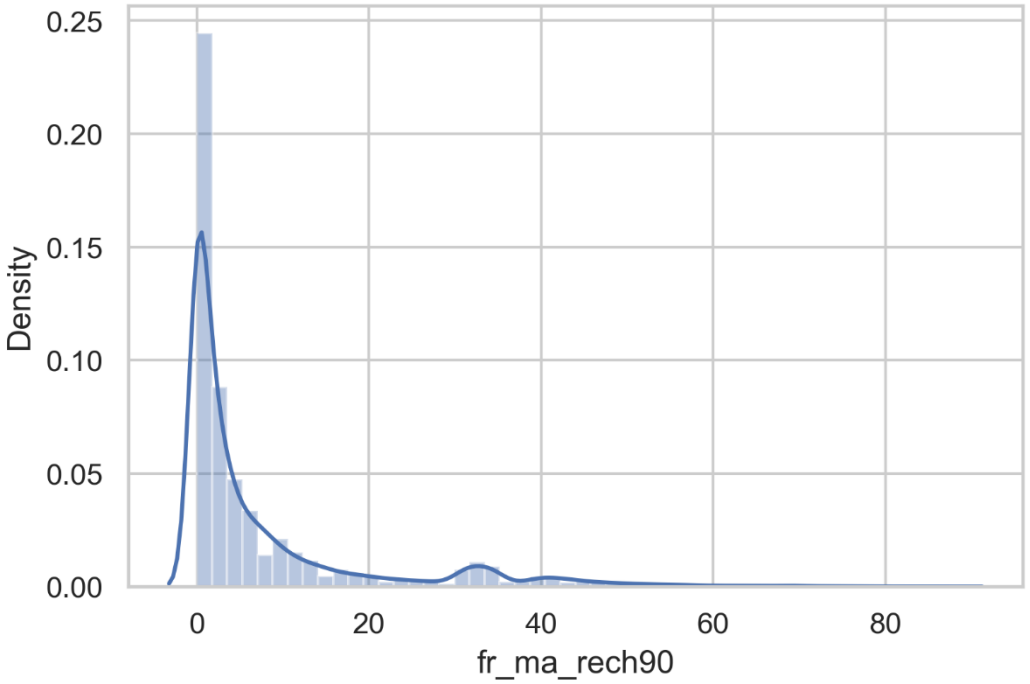

1e−5
8
7
6
5
4
3
2
1
0
Density
0.0
0.2
0.4
0.6
0.8
1.0
1e6
fr_ma_rech30

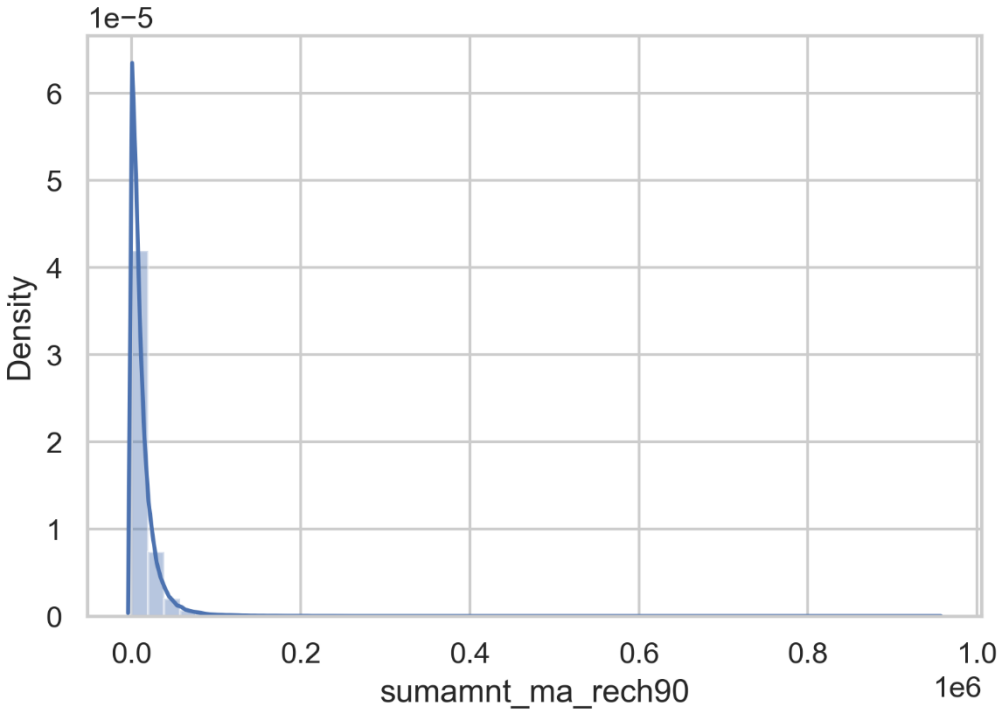

0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
Density
0
100000
200000
300000
400000
500000
600000
700000
800000
sumamnt_ma_rech30

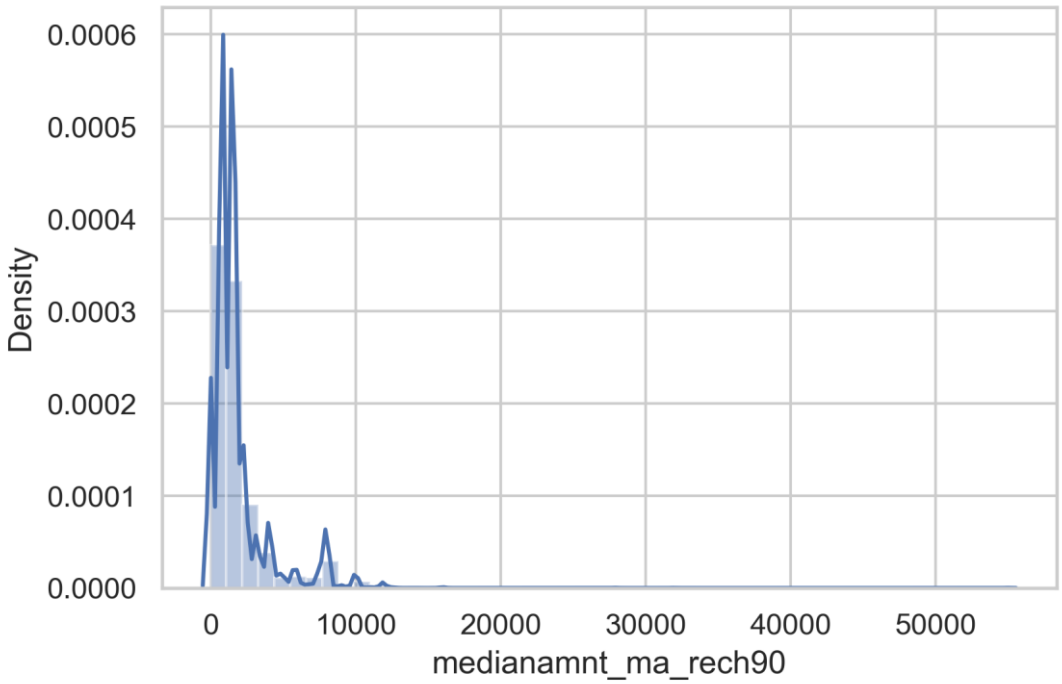

0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
Density
0
10000
20000
30000
40000
50000
medianamnt_ma_rech30

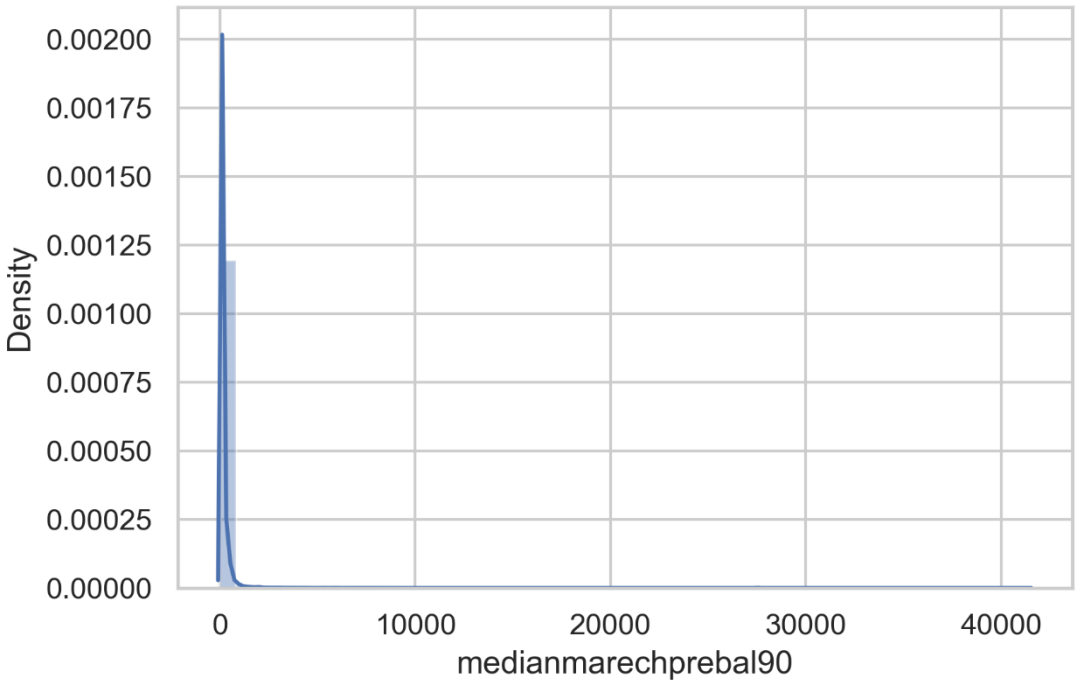

1e−5
8
7
6
5
4
3
2
1
0
Density
0.0
0.2
0.4
0.6
0.8
1.0
1e6
medianmarechprebal30

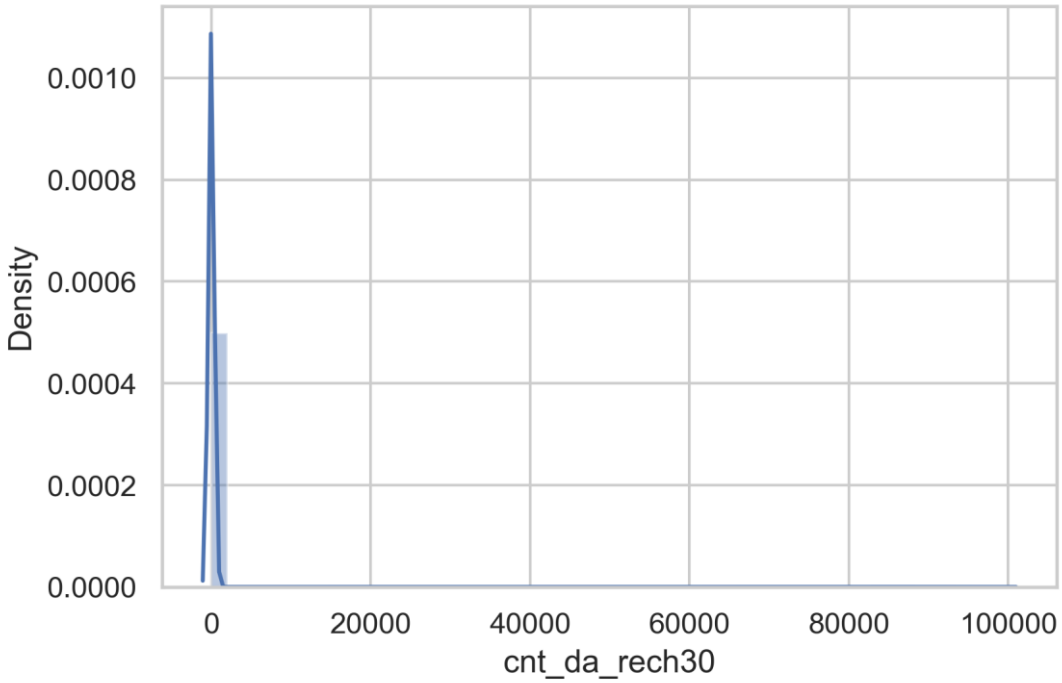

0.12
0.10
0.08
0.06
0.04
0.02
0.00
Density
0
50
100
150
200
250
300
350
cnt_ma_rech90

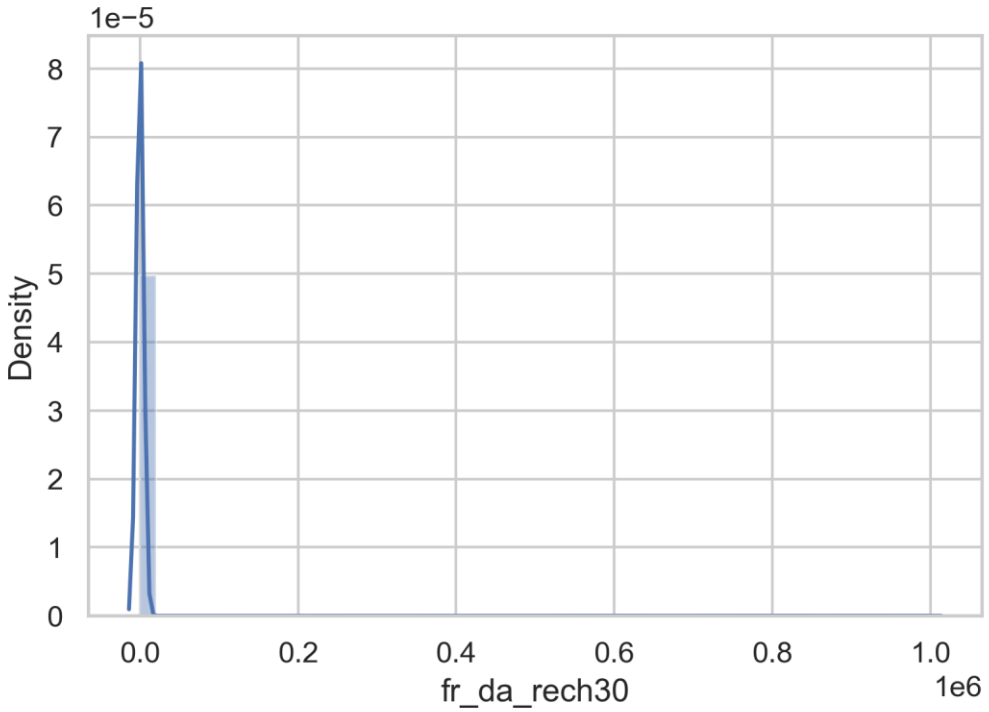

0.25
0.20
0.15
0.10
0.05
0.00
Density
0
20
40
60
80
fr_ma_rech90

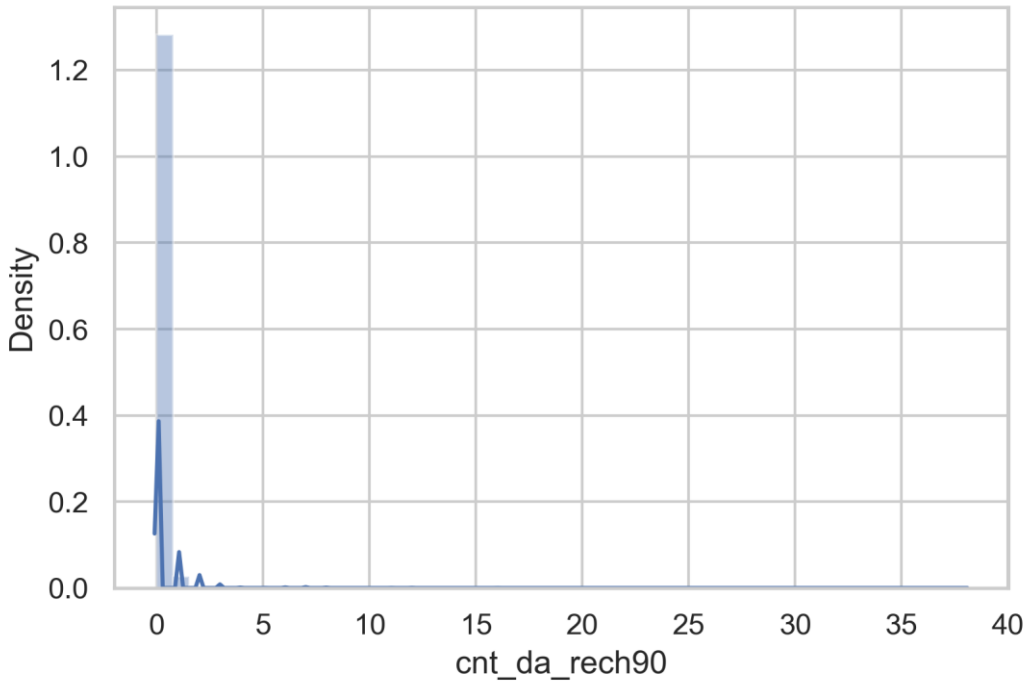

1e−5
6
5
4
3
2
1
0
Density
0.0
0.2
0.4
0.6
0.8
1.0
1e6
sumamnt_ma_rech90

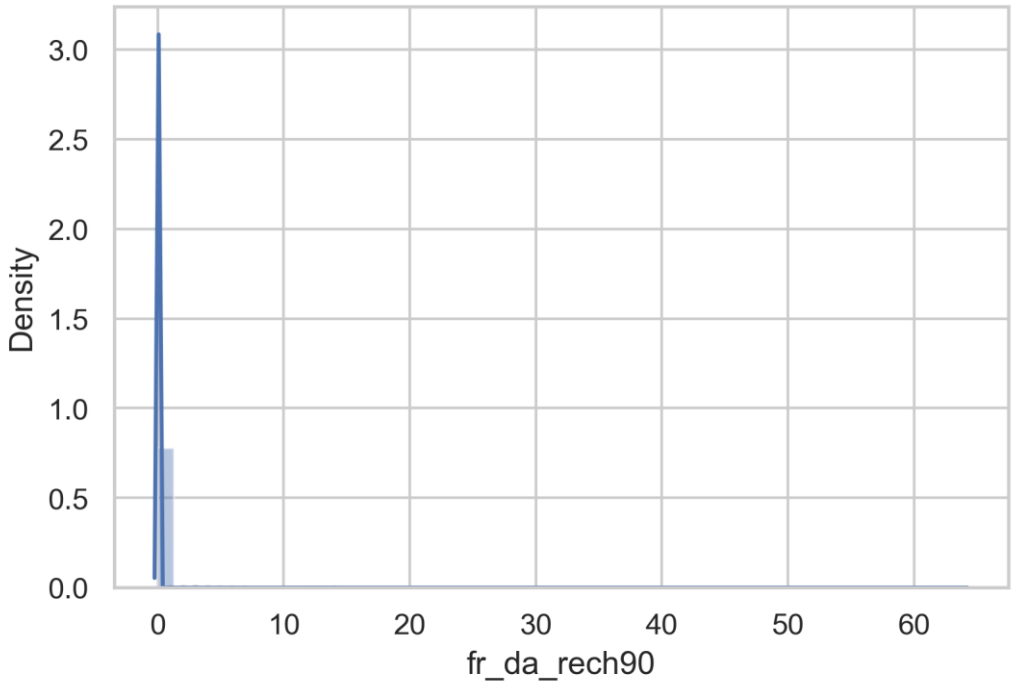

0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
Density
0
10000
20000
30000
40000
50000
medianamnt_ma_rech90

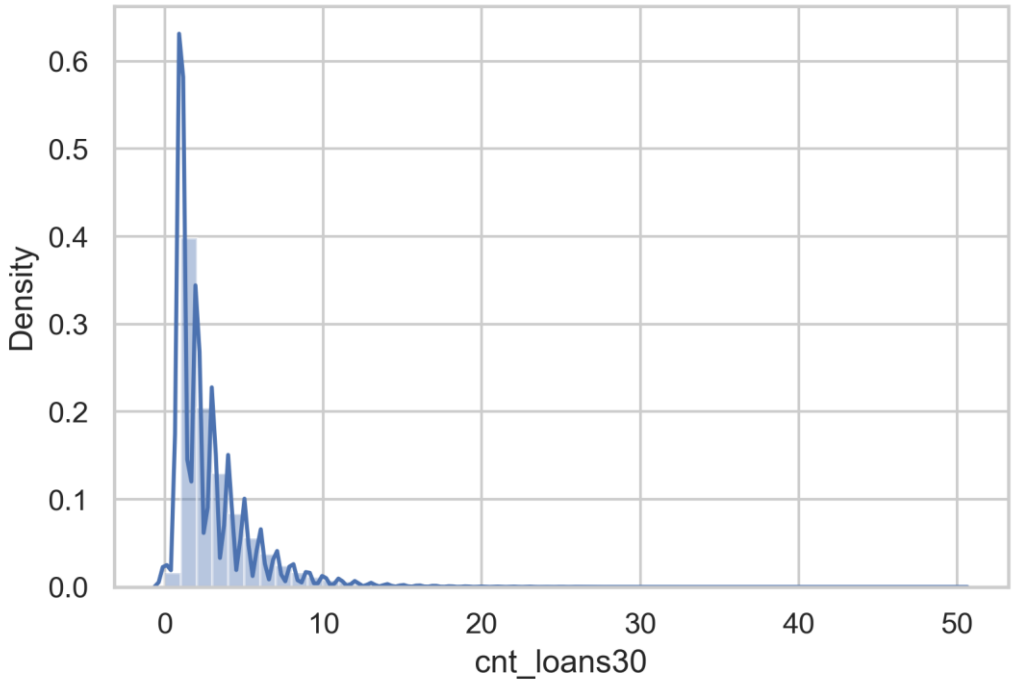

0.00200
0.00175
0.00150
0.00125
0.00100
0.00075
0.00050
0.00025
0.00000
Density
0
10000
20000
30000
40000
medianmarechprebal90

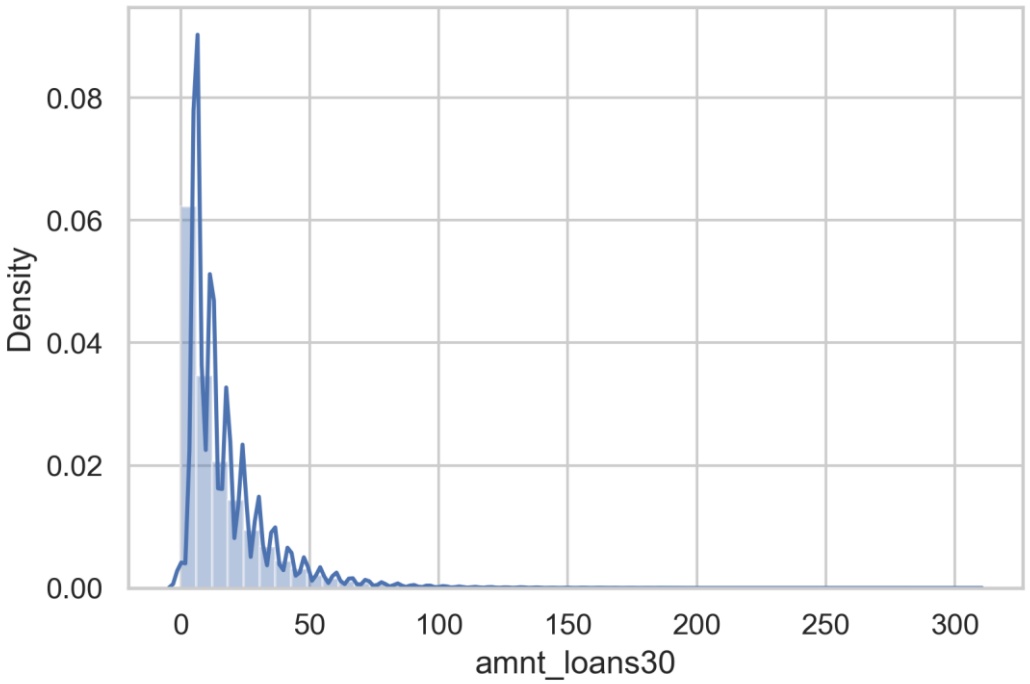

0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
Density
0
20000
40000
60000
80000
100000
cnt_da_rech30

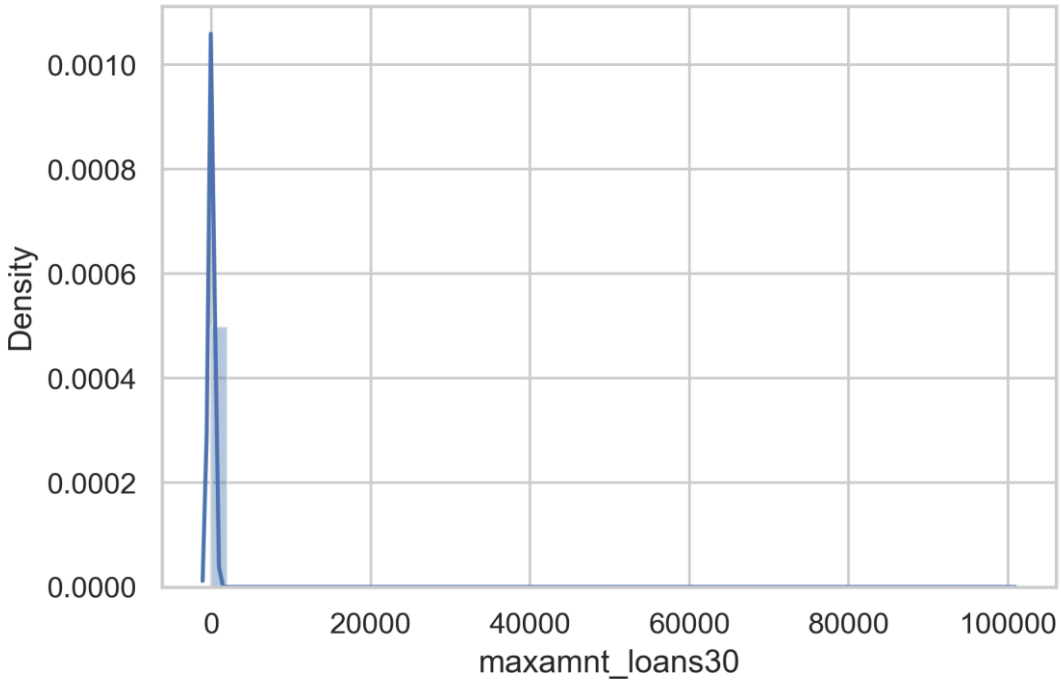

1e−5
8
7
6
5
4
3
2
1
0
Density
0.0
0.2
0.4
0.6
0.8
1.0
1e6
fr_da_rech30

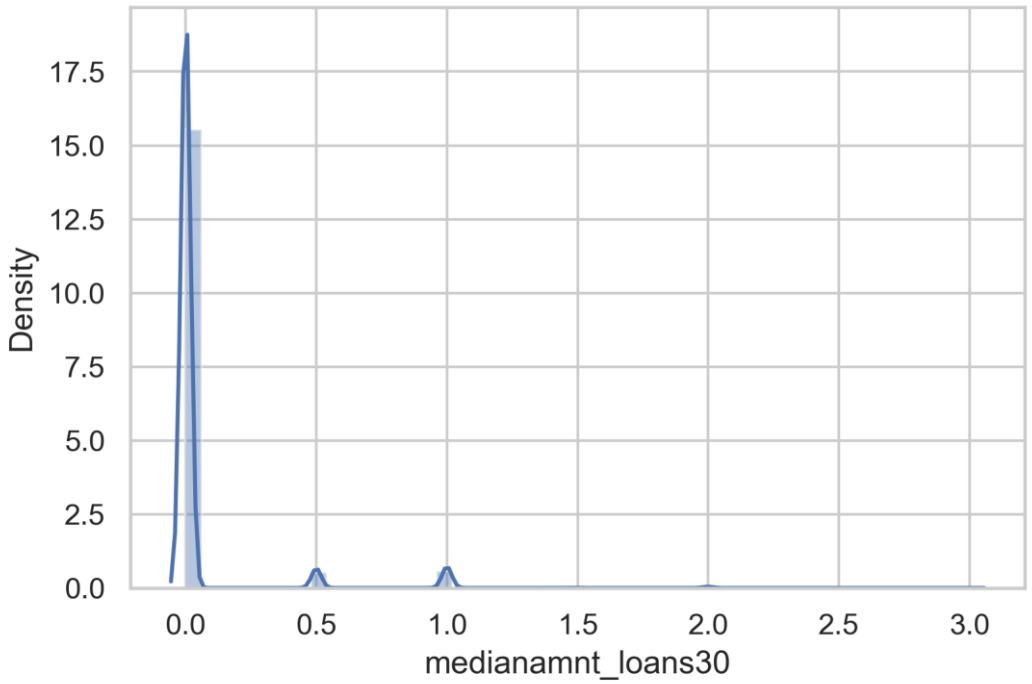

1.2
1.0
0.8
0.6
0.4
0.2
0.0
Density
0
5
10
15
20
25
30
35
40
cnt_da_rech90

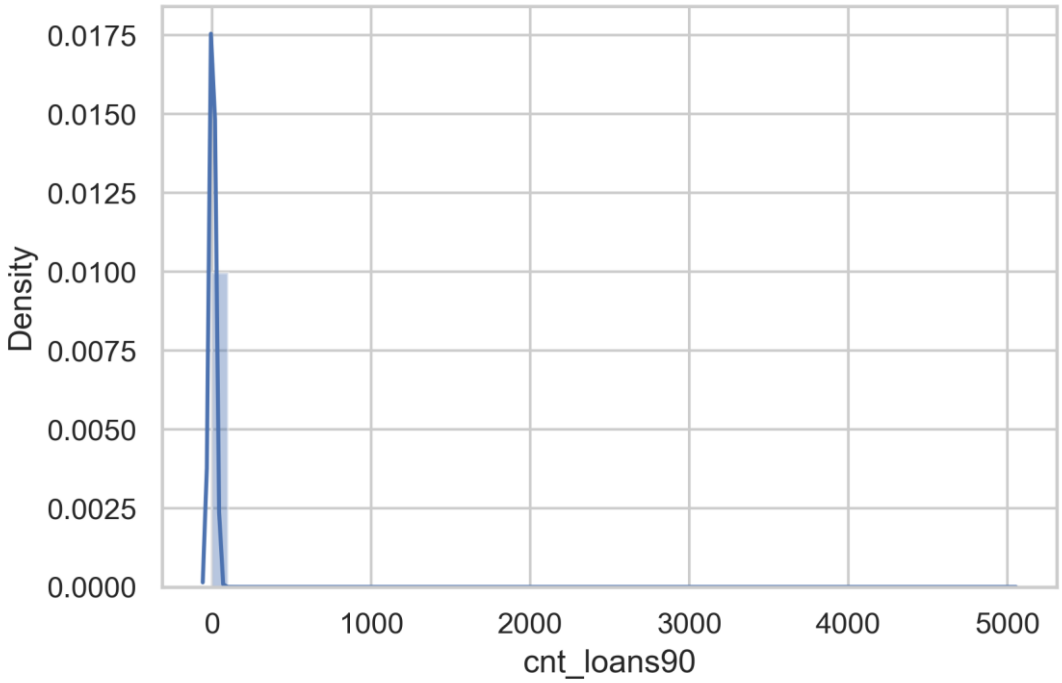

Density
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
10
20
30
40
50
60
fr_da_rech90

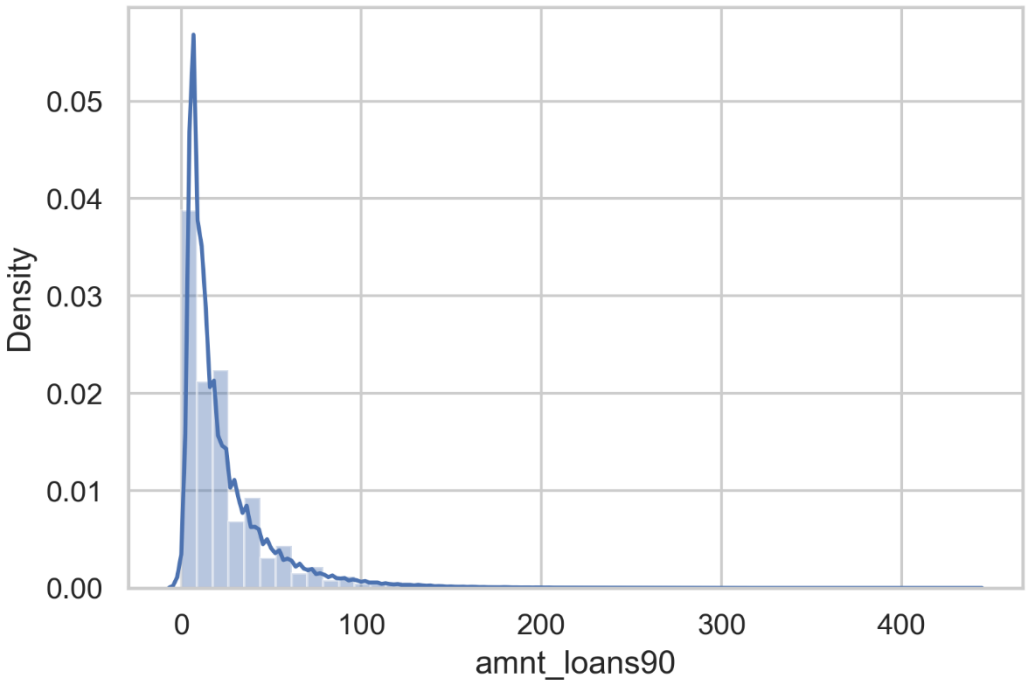

Density
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40
50
cnt_loans30

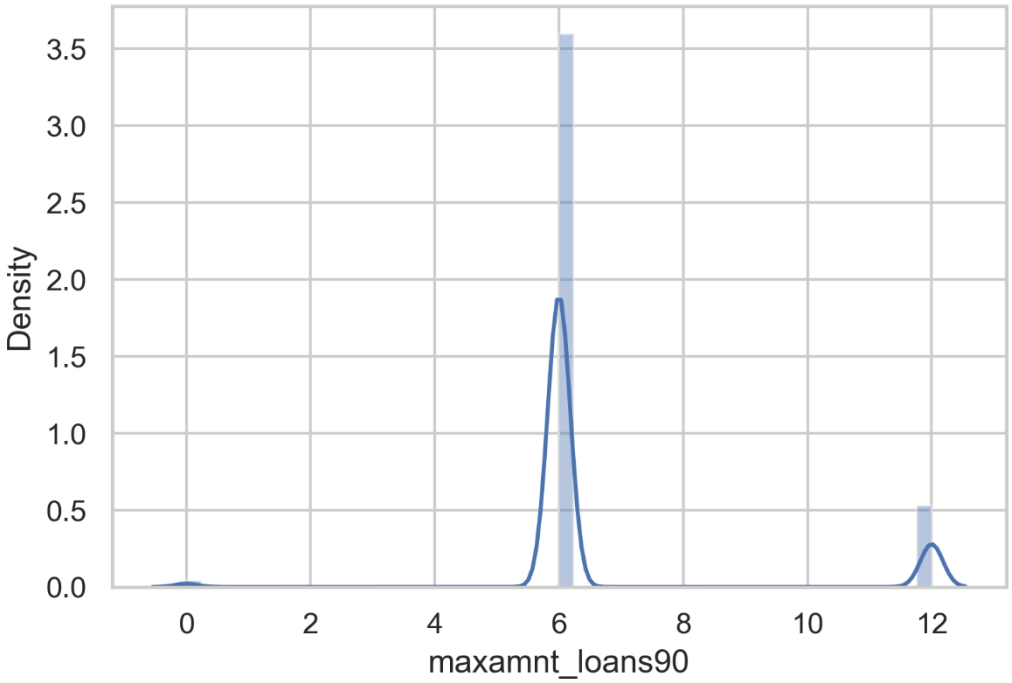

0.08
0.06
0.04
0.02
0.00
Density
0
50
100
150
200
250
300
amnt_loans30

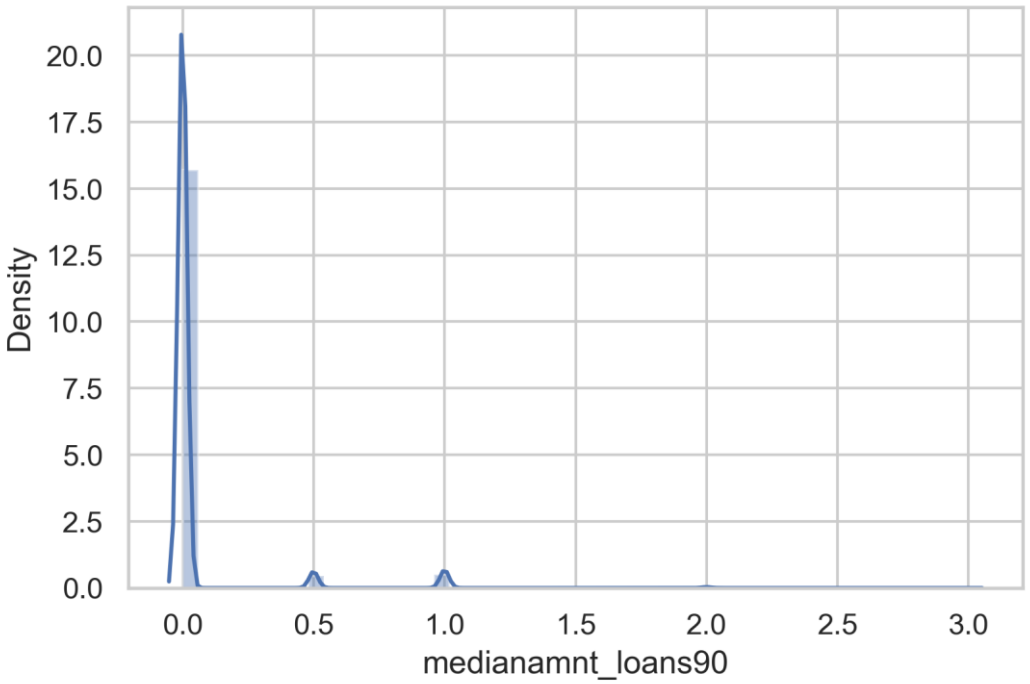

0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
Density
0
20000
40000
60000
80000
100000
maxamnt_loans30

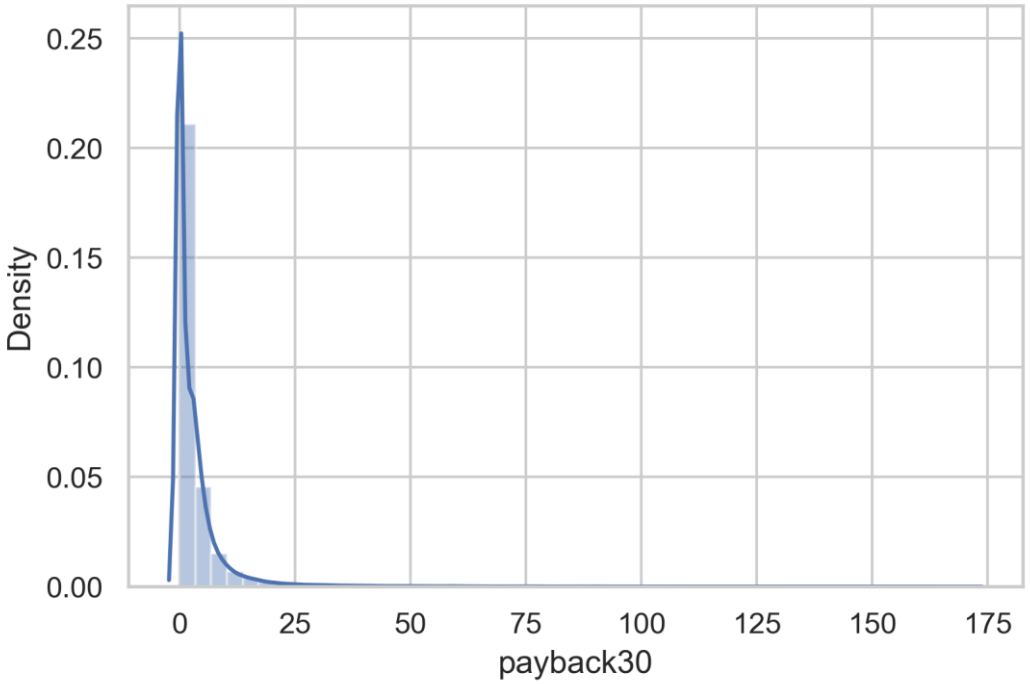

Density
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
medianamnt_loans30

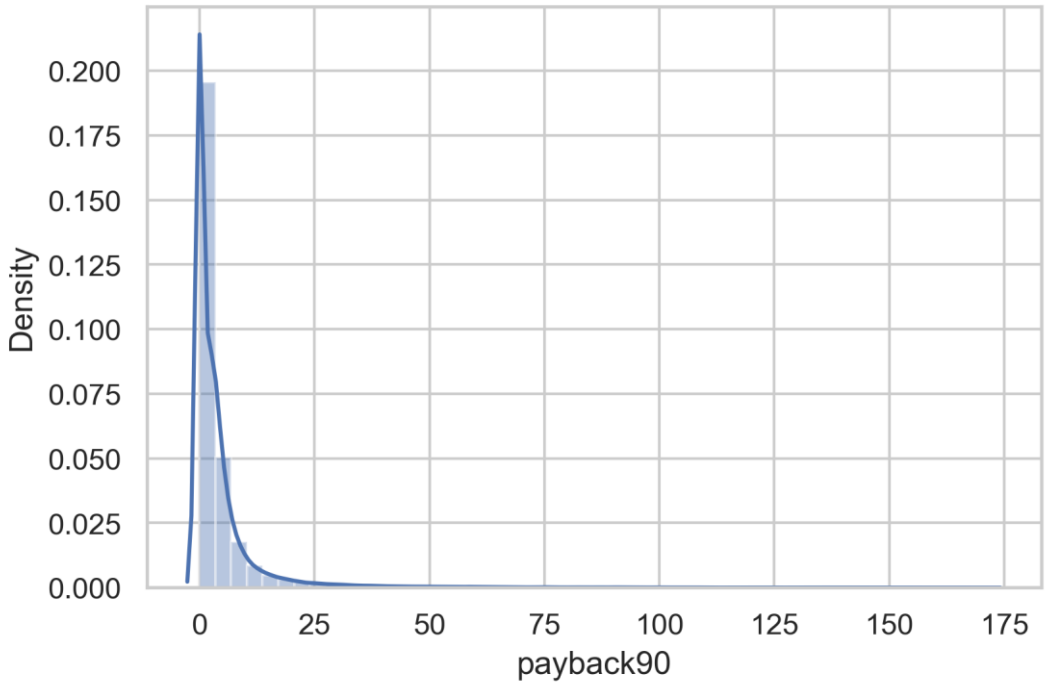

Density
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
1000
2000
3000
4000
5000
cnt_loans90

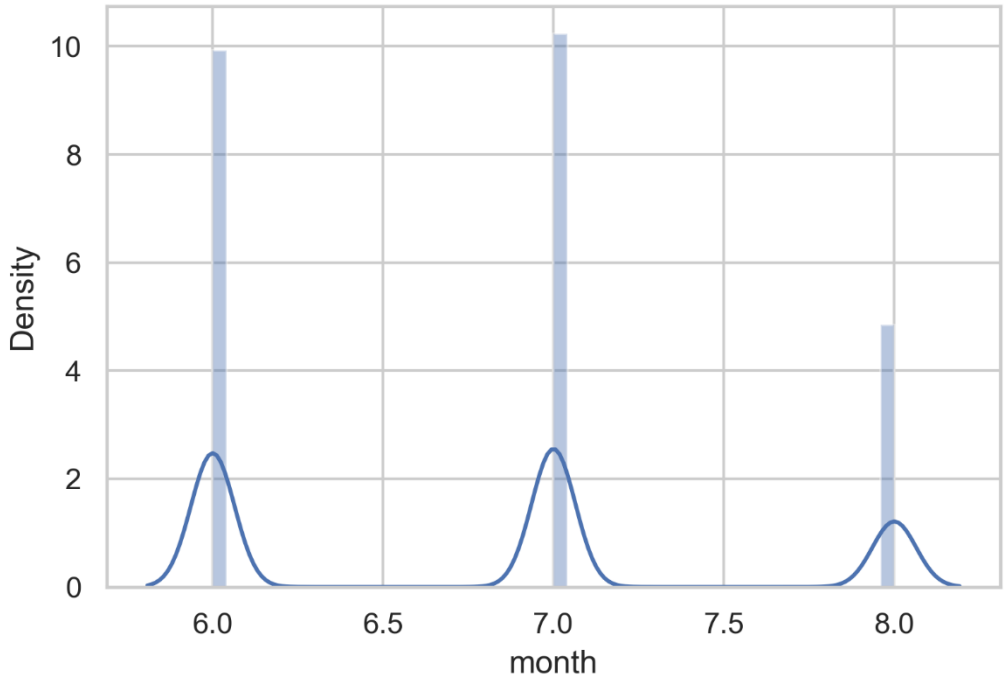

0.05
0.04
0.03
0.02
0.01
0.00
Density
0
100
200
300
400
amnt_loans90

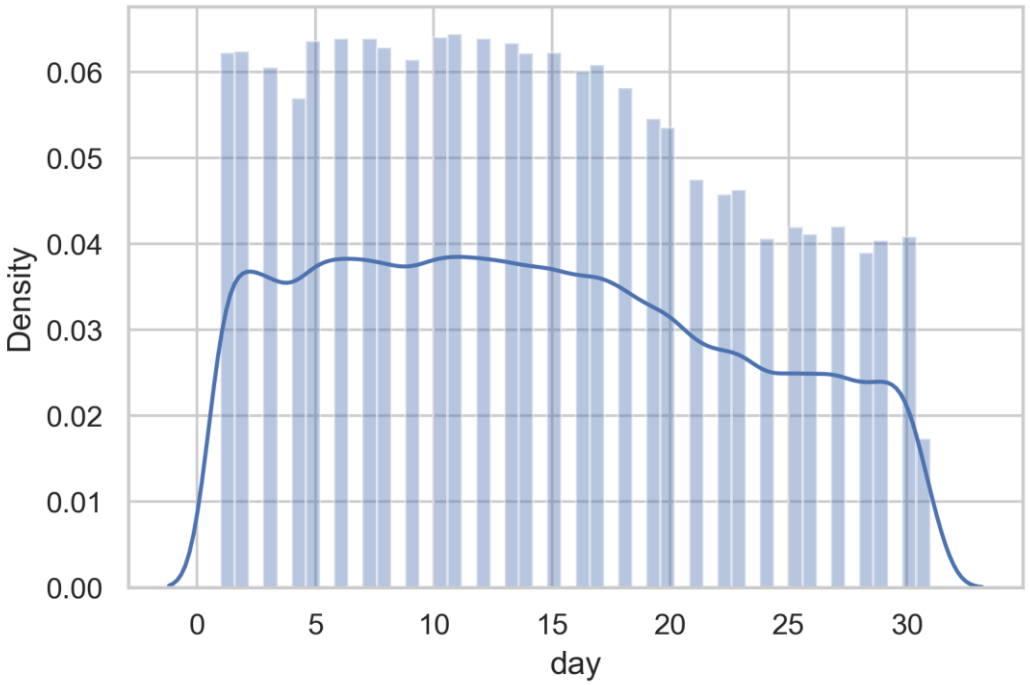

3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Density
0
2
4
6
8
10
12
maxamnt_loans90

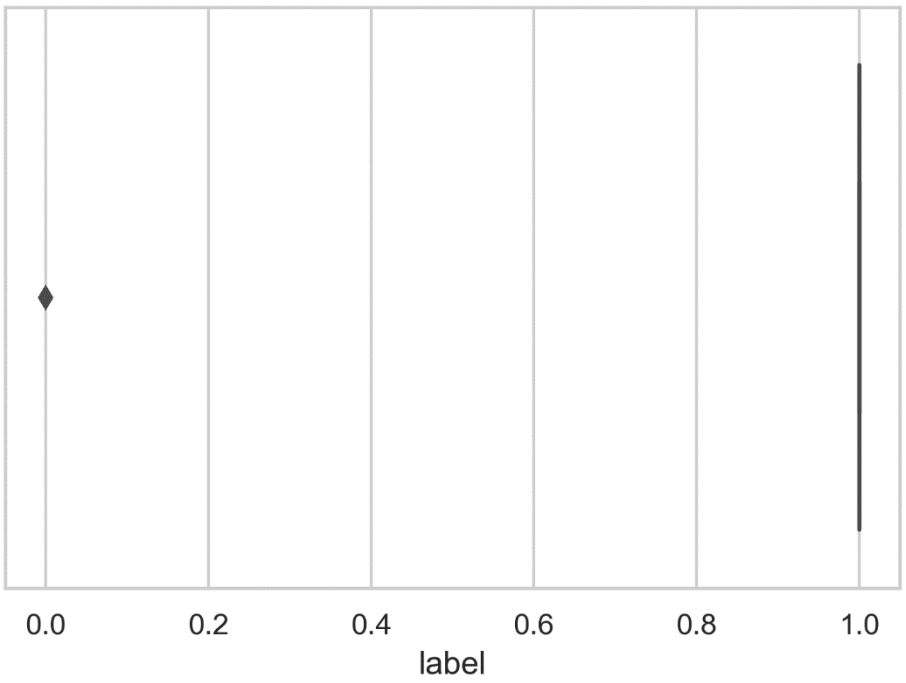

Density
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
medianamnt_loans90

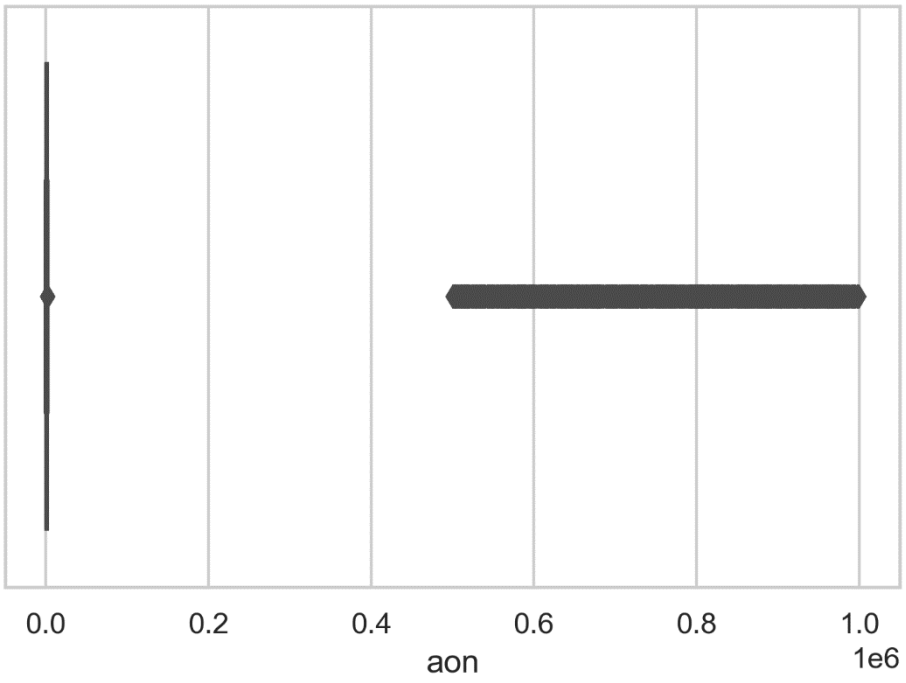

Density
0.25
0.20
0.15
0.10
0.05
0.00
0
25
50
75
100
125
150
175
payback30

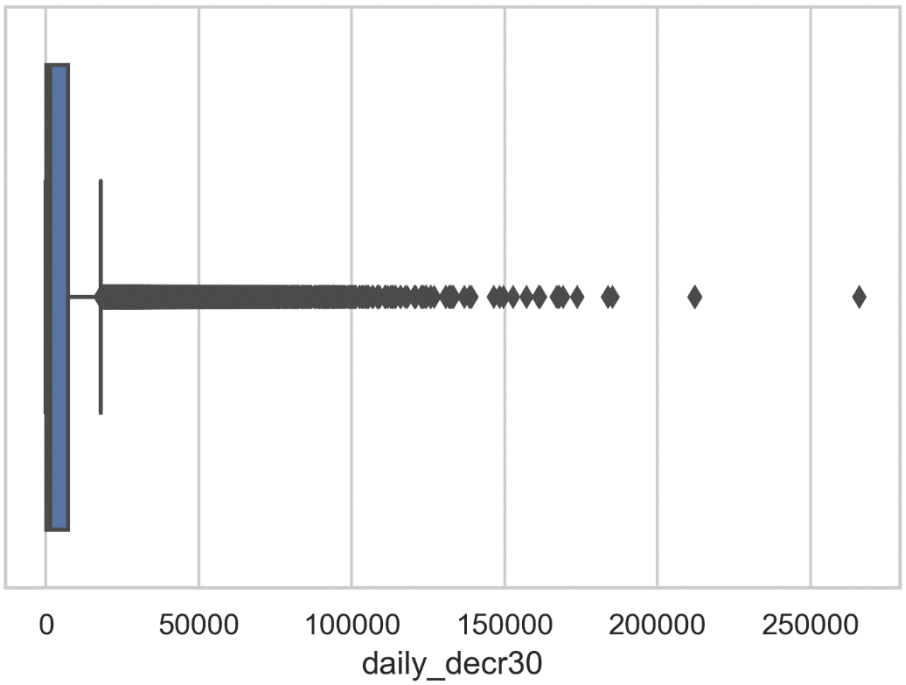

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Density
0
25
50
75
100
125
150
175
payback90

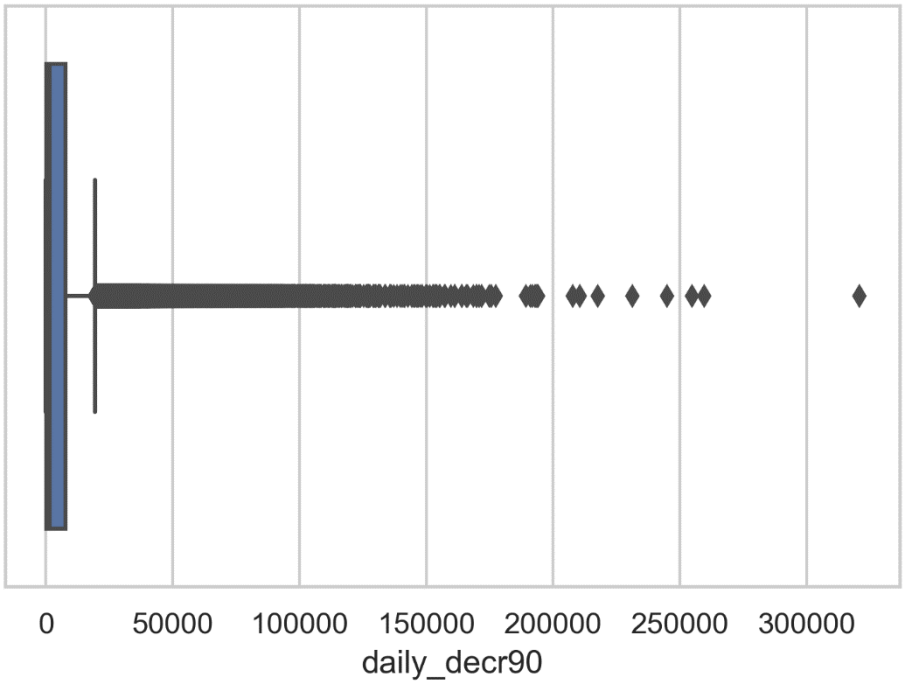

10
8
6
4
2
0
Density
6.0
6.5
7.0
7.5
8.0
month

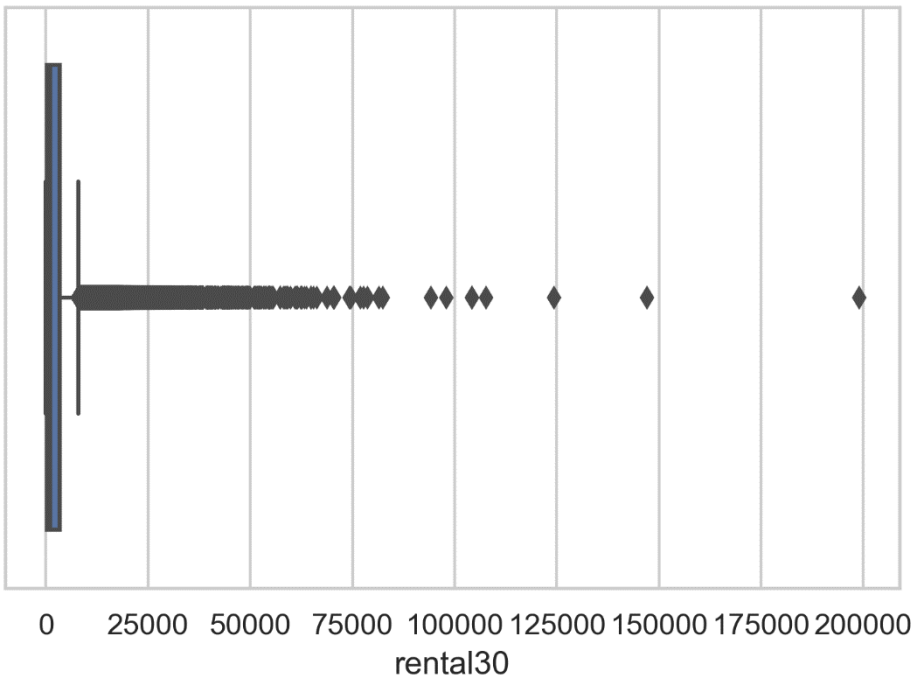

0.06
0.05
0.04
0.03
0.02
0.01
0.00
Density
0
5
10
15
20
25
30
day

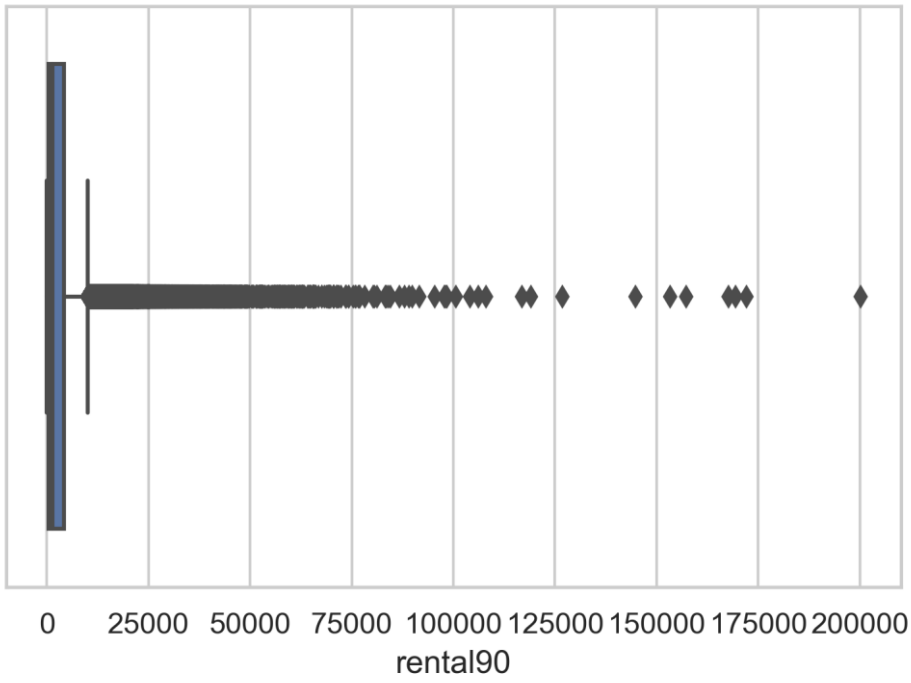

0.0
0.2
0.4
0.6
0.8
1.0
label

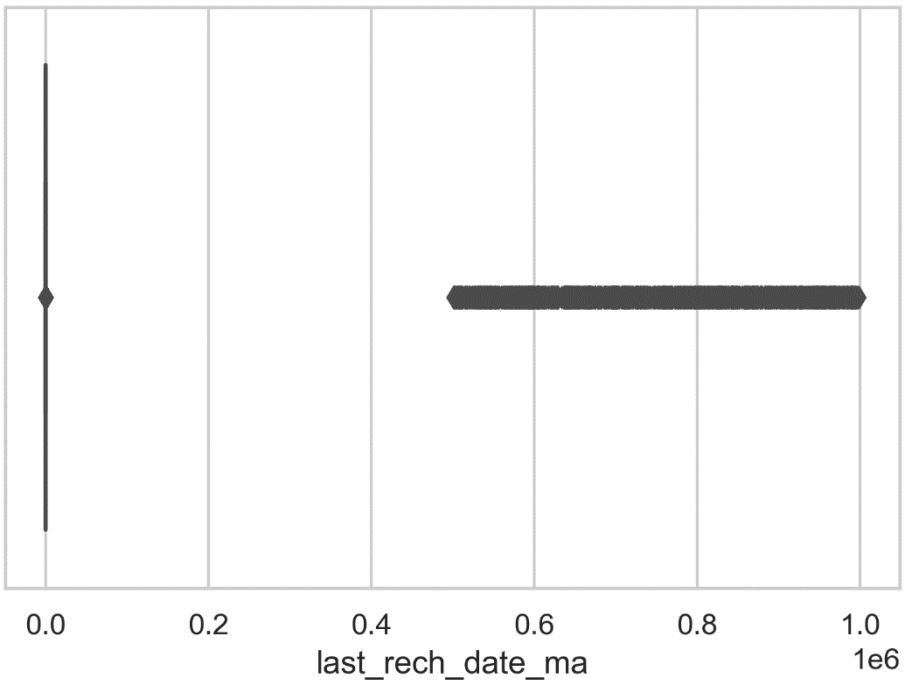

0.0
0.2
0.4
0.6
0.8
1.0
1e6
aon

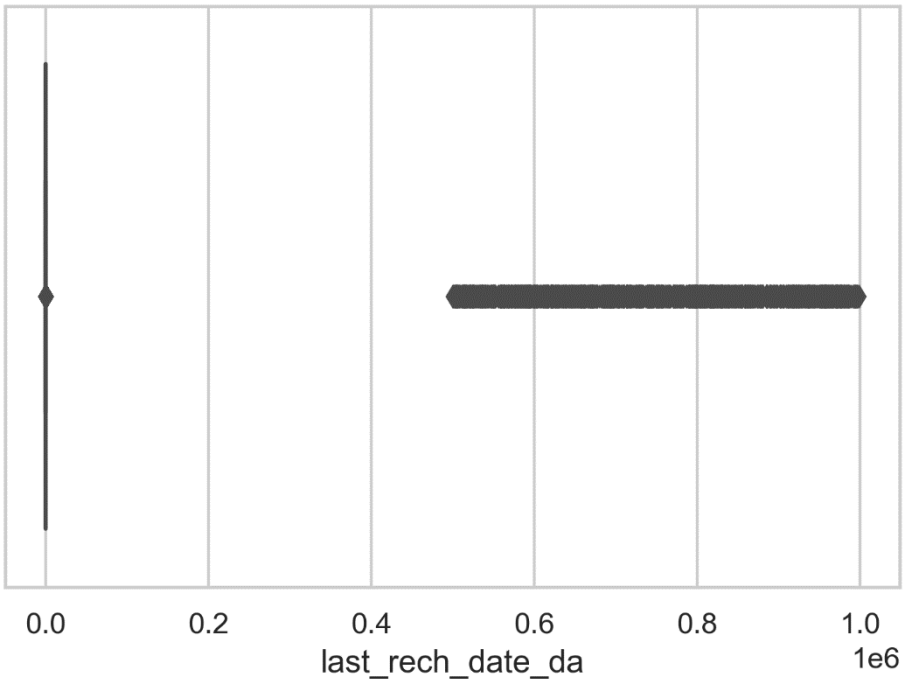

0
50000
100000
150000
200000
250000
daily_decr30

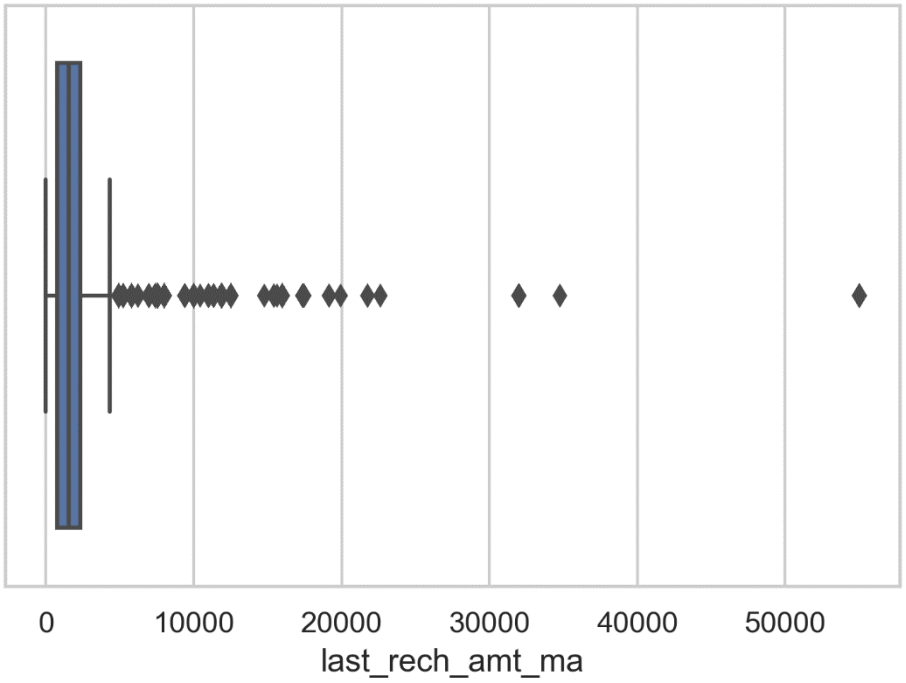

0
50000
100000
150000
200000
250000
300000
daily_decr90

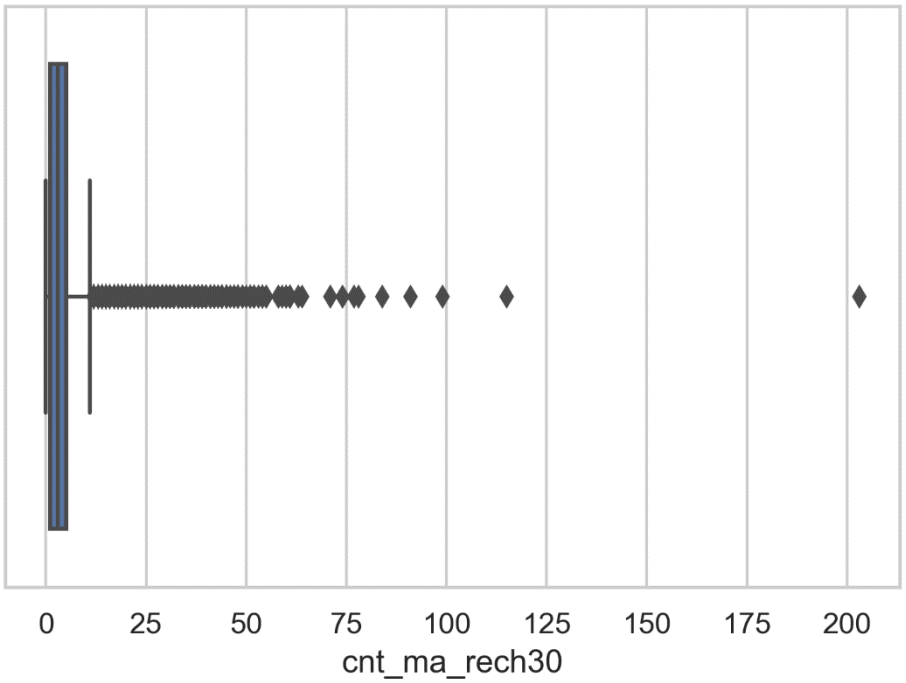

0
25000
50000
75000
100000
125000
150000
175000
200000
rental30

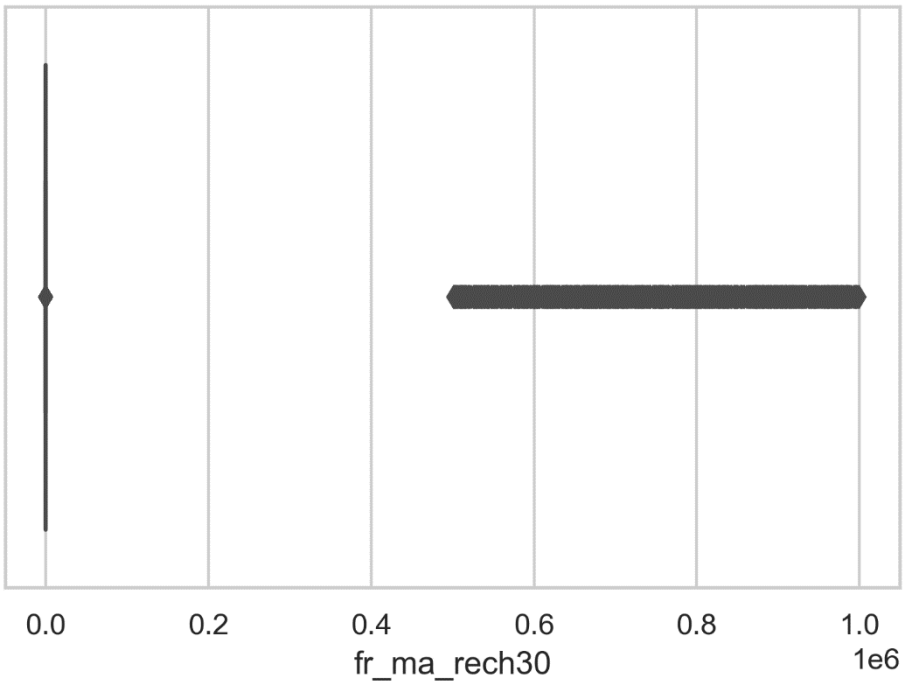

0
25000
50000
75000
100000
125000
150000
175000
200000
rental90

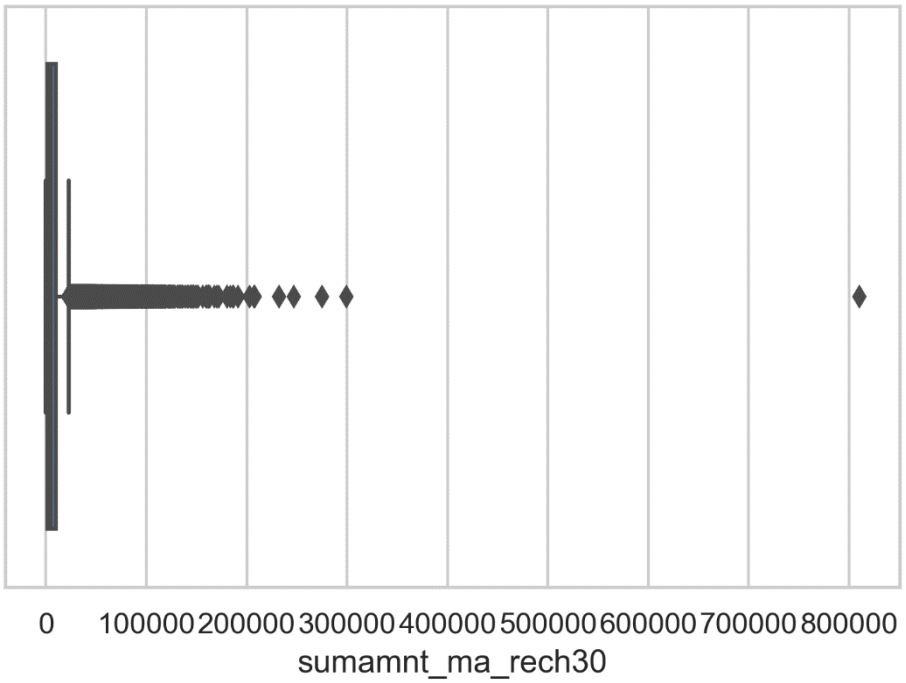

0.0
0.2
0.4
0.6
0.8
1.0
1e6
last_rech_date_ma

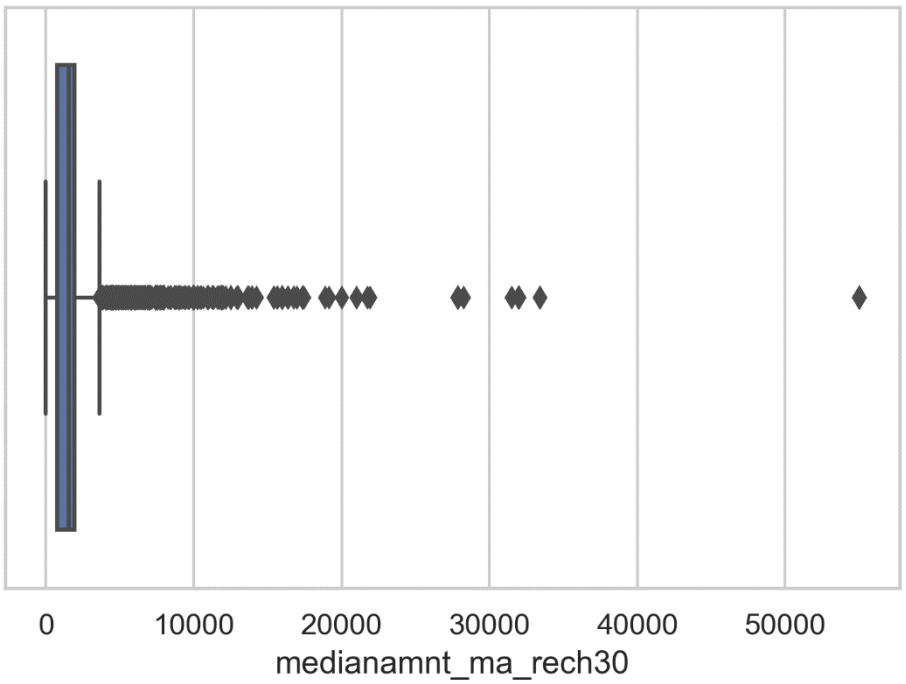

0.0
0.2
0.4
0.6
0.8
1.0
1e6
last_rech_date_da

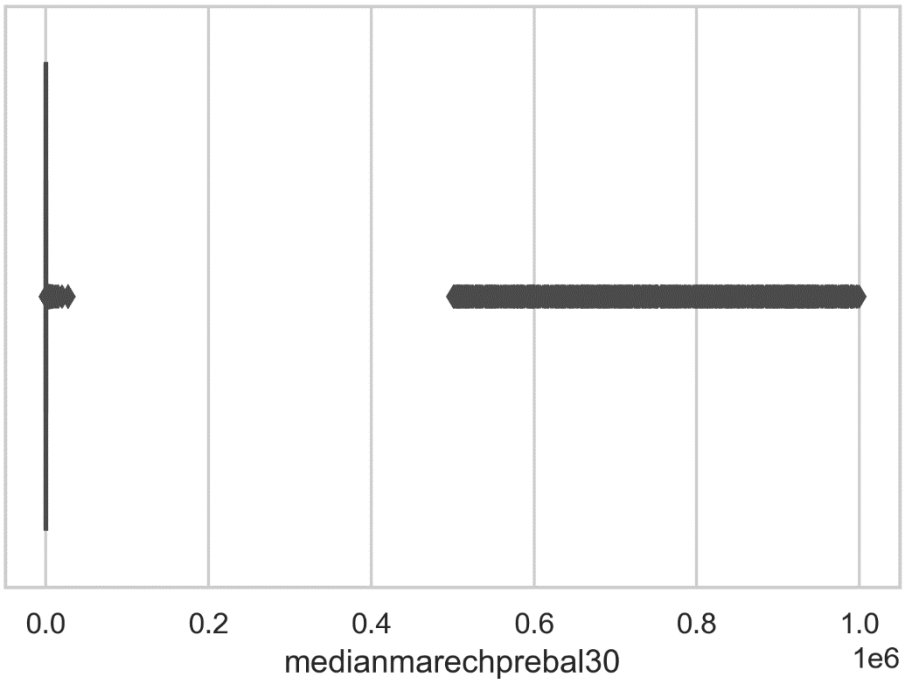

0
10000
20000
30000
40000
50000
last_rech_amt_ma

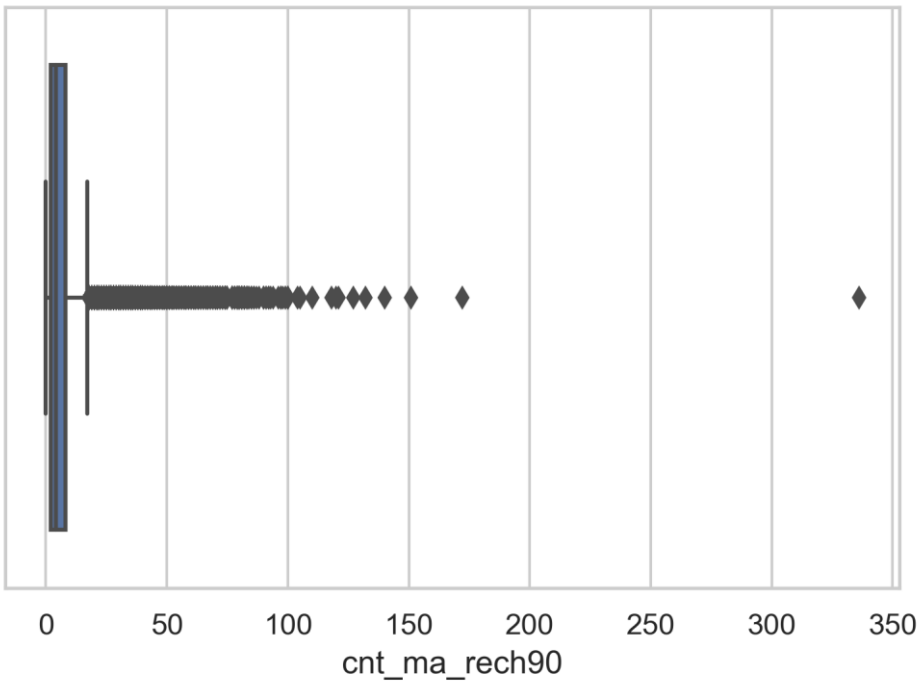

0
25
50
75
100
125
150
175
200
cnt_ma_rech30

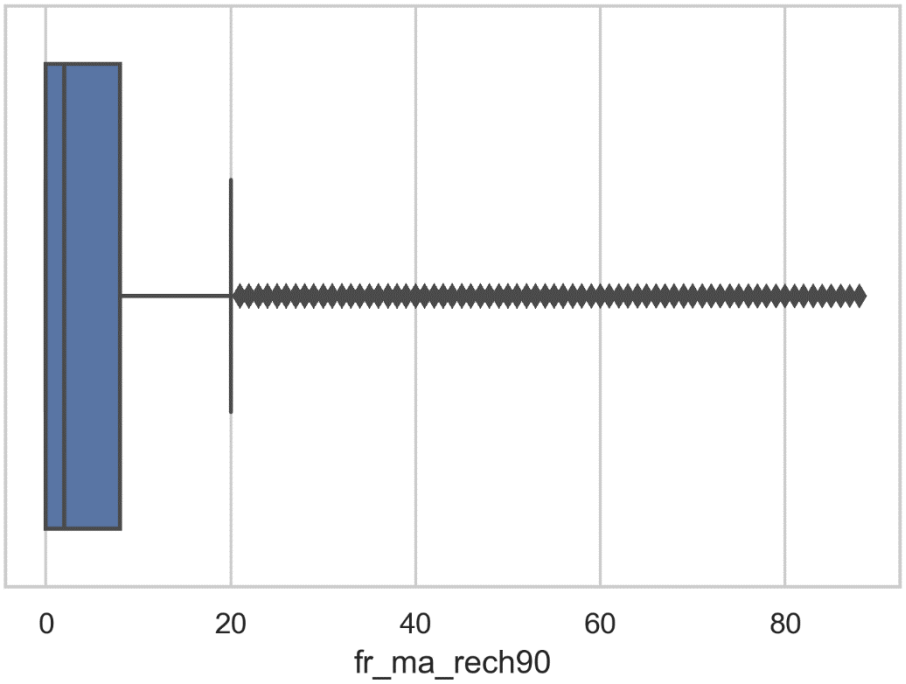

0.0
0.2
0.4
0.6
0.8
1.0
1e6
fr_ma_rech30

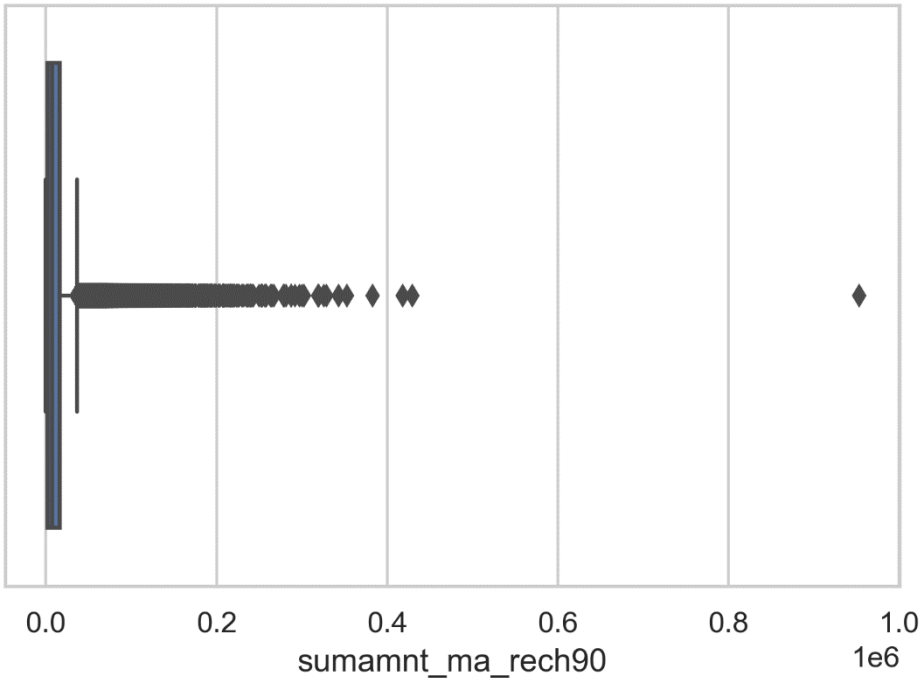

0
100000
200000
300000
400000
500000
600000
700000
800000
sumamnt_ma_rech30

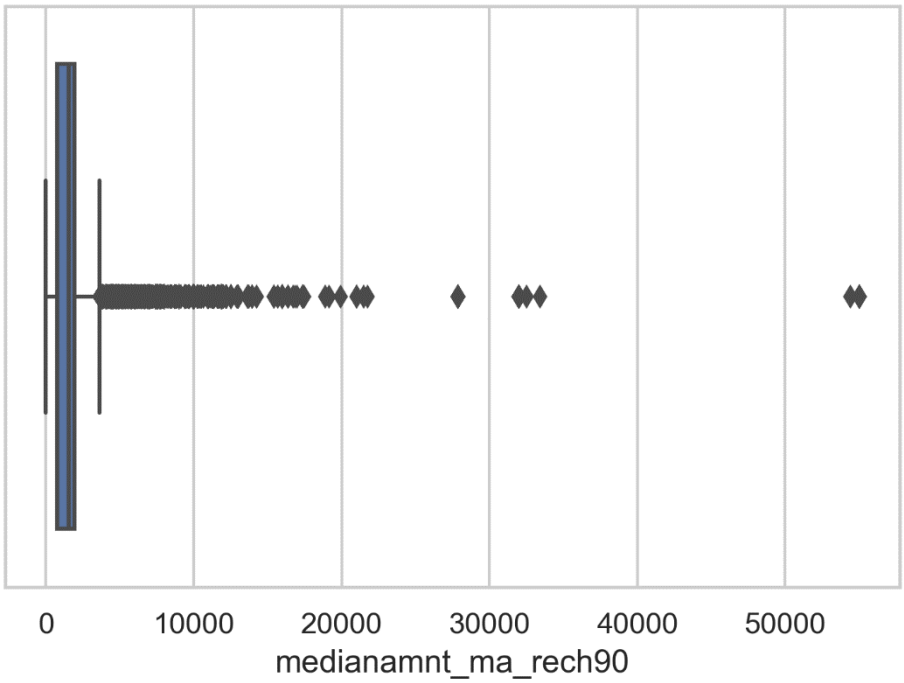

0
10000
20000
30000
40000
50000
medianamnt_ma_rech30

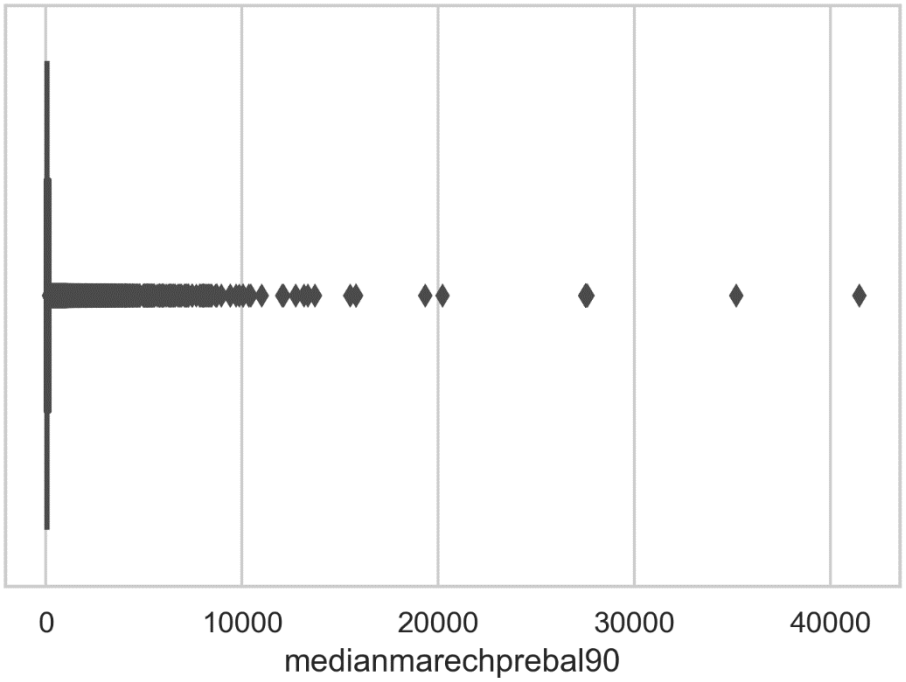

0.0
0.2
0.4
0.6
0.8
1.0
1e6
medianmarechprebal30

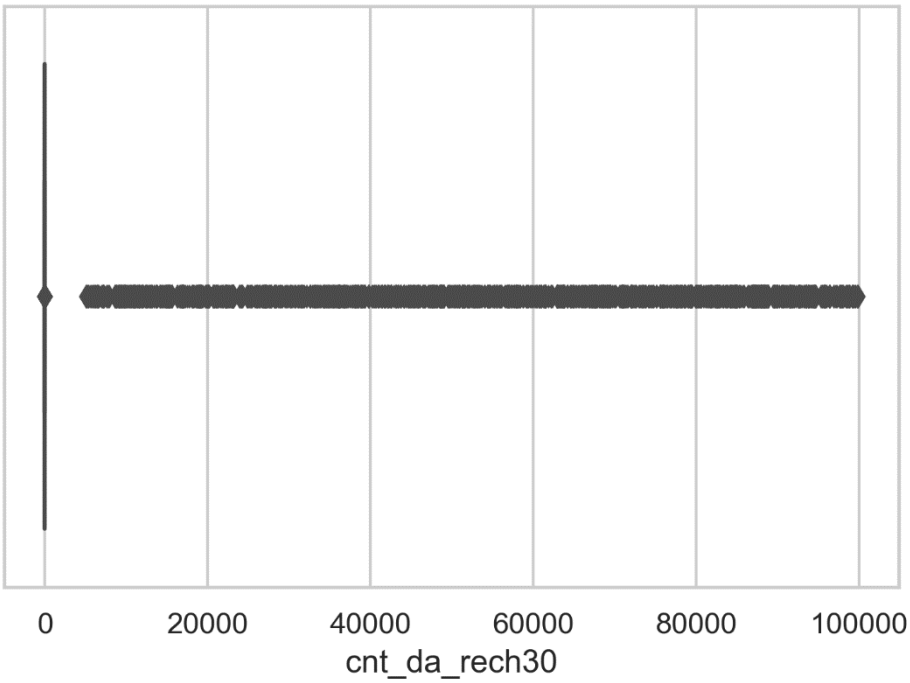

0
50
100
150
200
250
300
350
cnt_ma_rech90

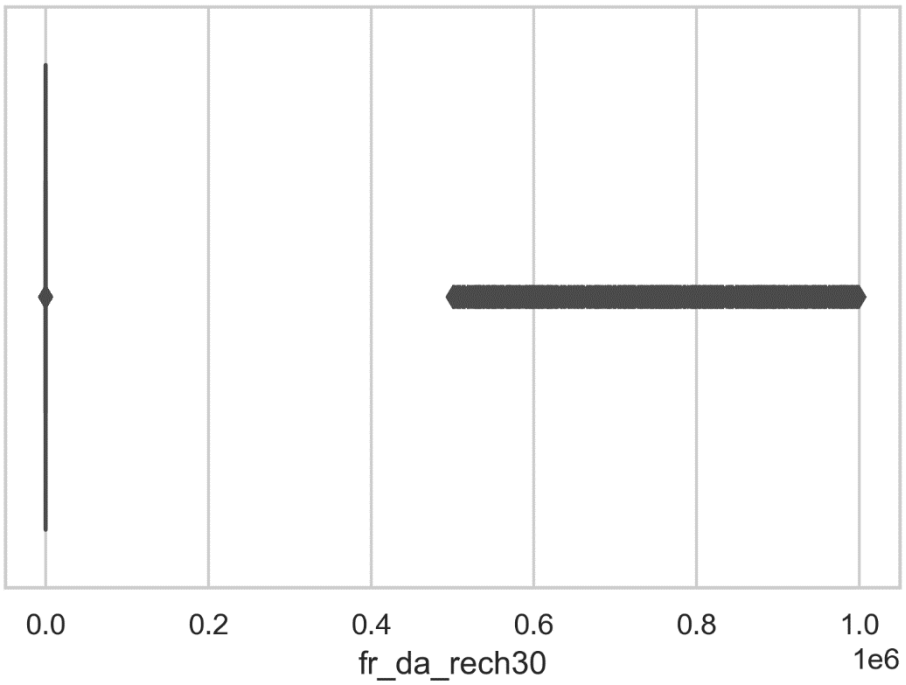

0
20
40
60
80
fr_ma_rech90

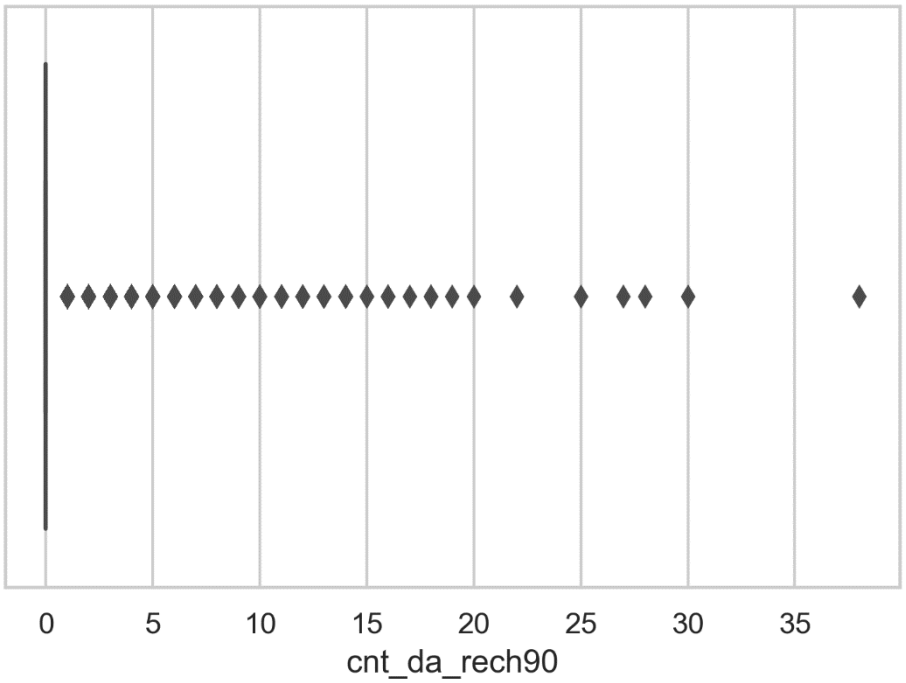

0.0
0.2
0.4
0.6
0.8
1.0
sumamnt_ma_rech90
1e6

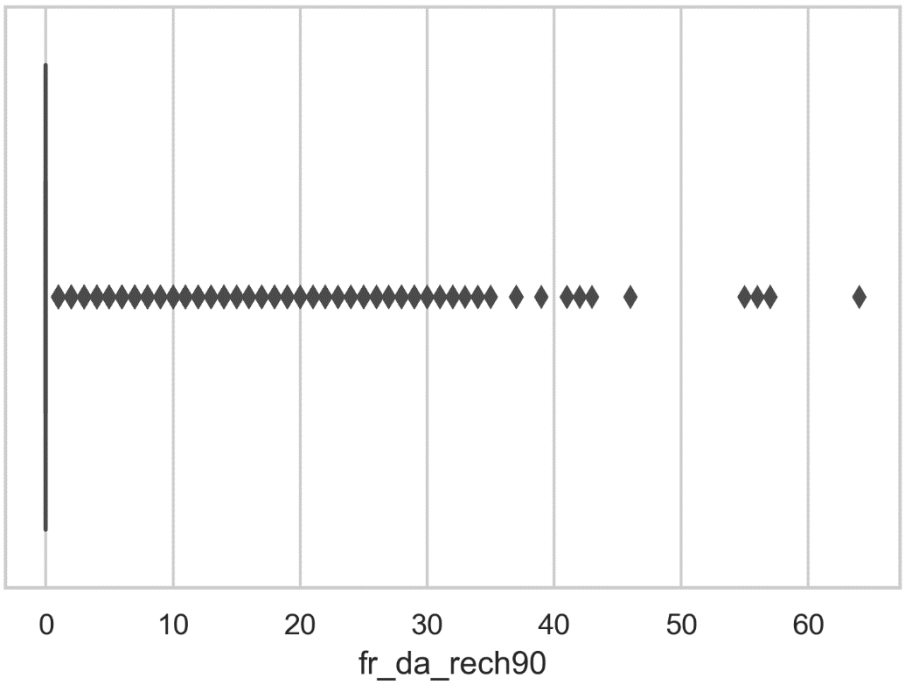

0
10000
20000
30000
40000
50000
medianamnt_ma_rech90

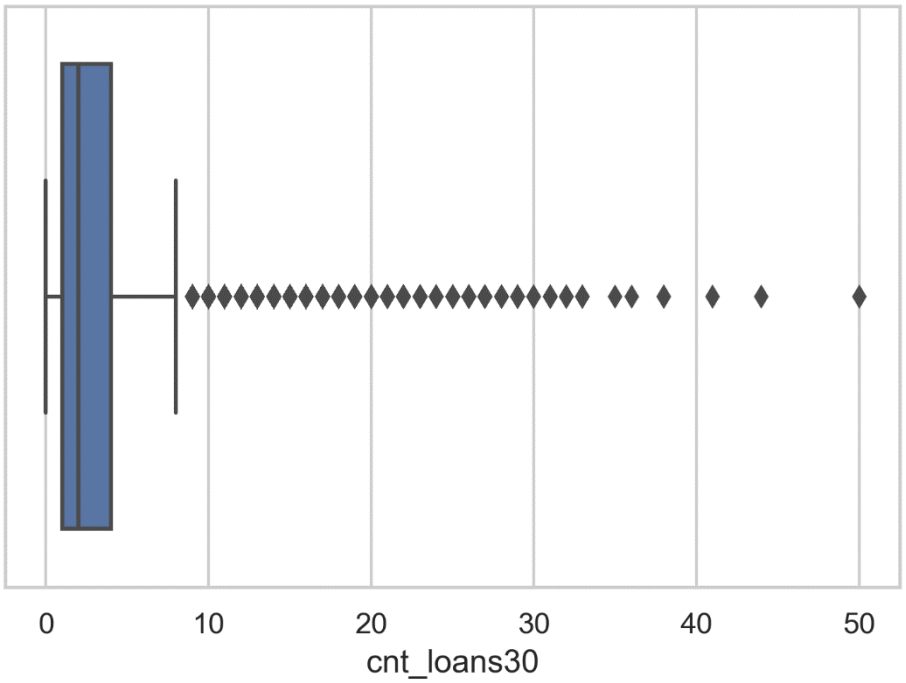

0
10000
20000
30000
40000
medianmarechprebal90

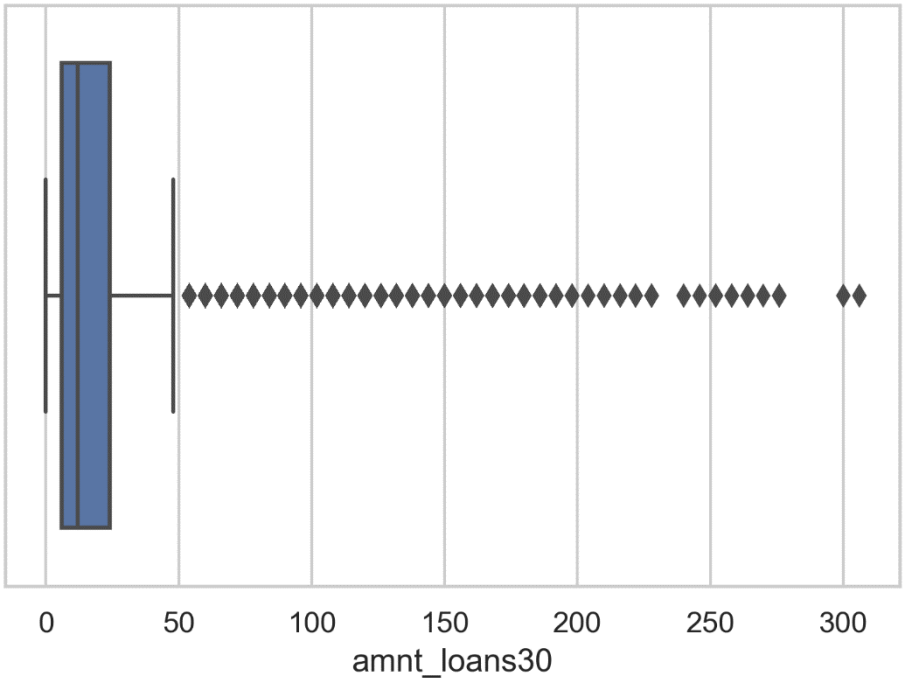

0
20000
40000
60000
80000
100000
cnt_da_rech30

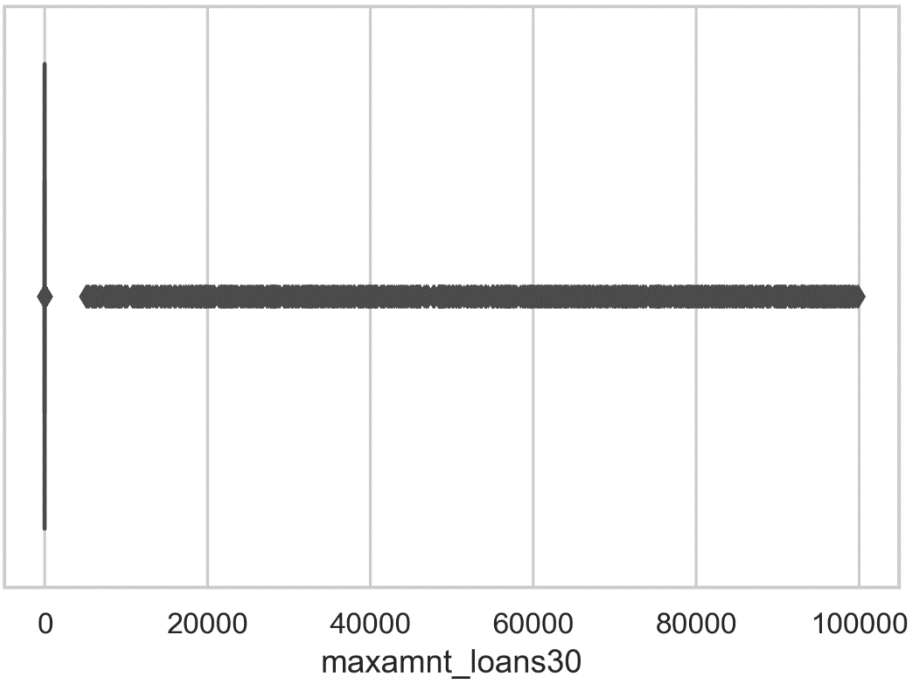

0.0
0.2
0.4
0.6
0.8
1.0
1e6
fr_da_rech30

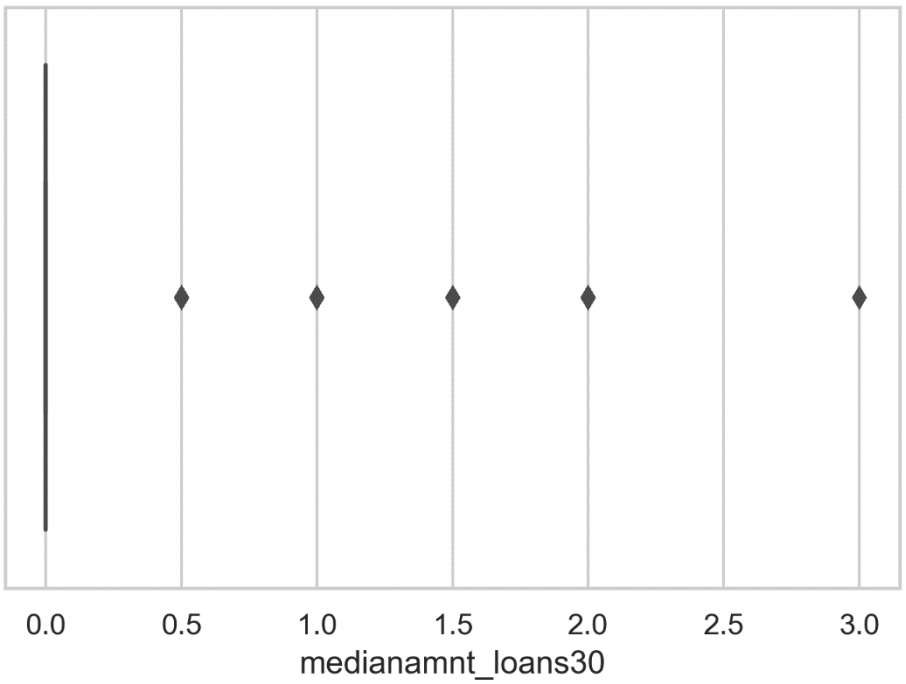

0
5
10
15
20
25
30
35
cnt_da_rech90

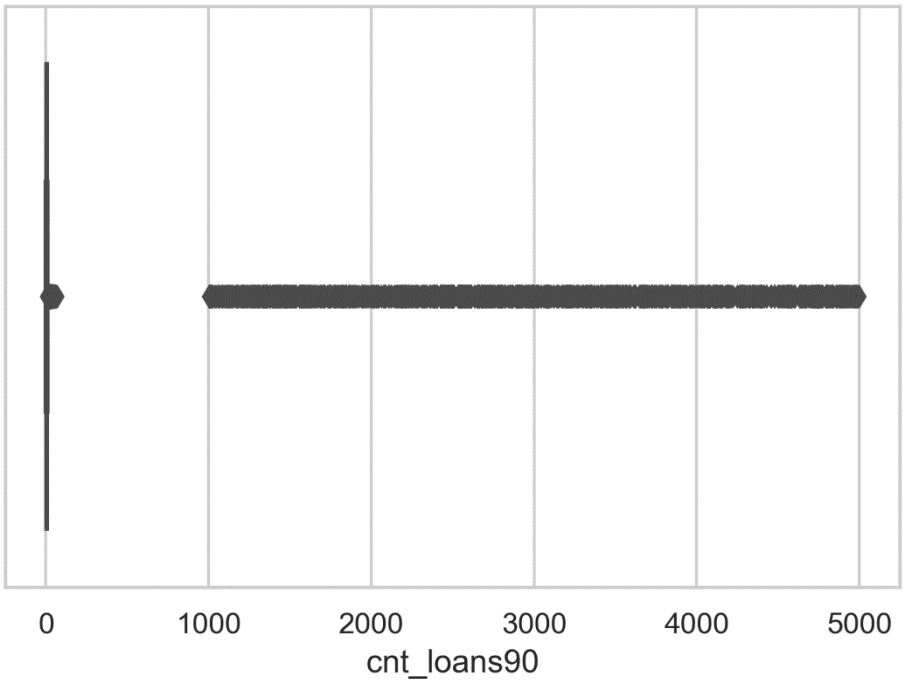

0
10
20
30
40
50
60
fr_da_rech90

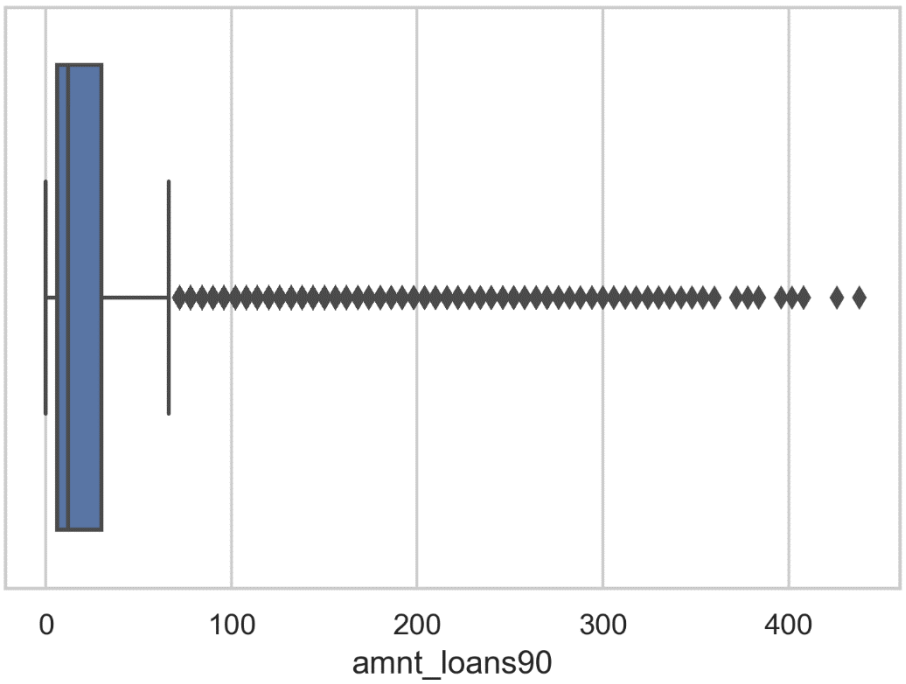

0
10
20
30
40
50
cnt_loans30

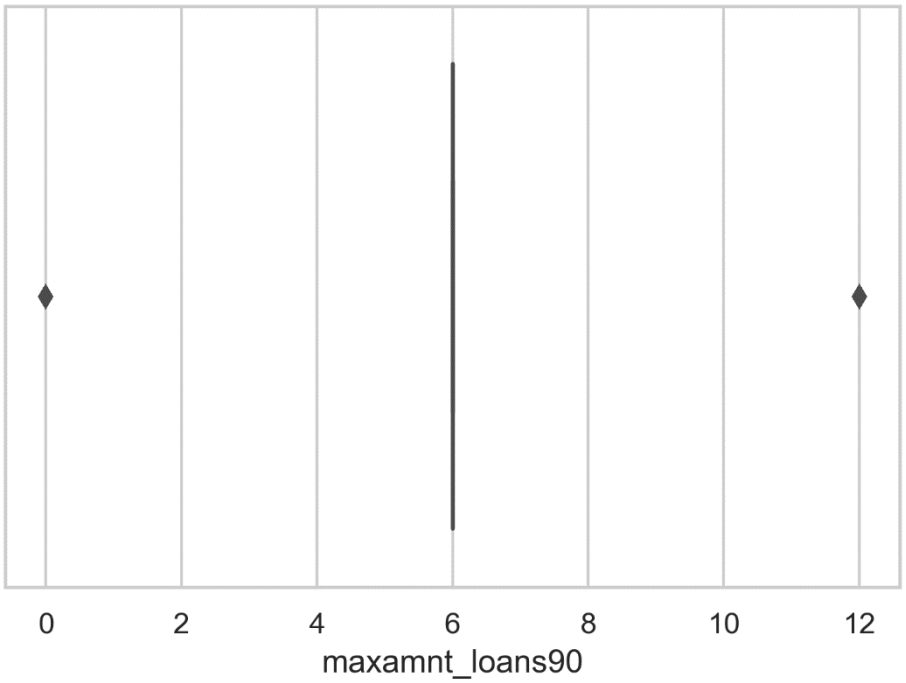

0
50
100
150
200
250
300
amnt_loans30

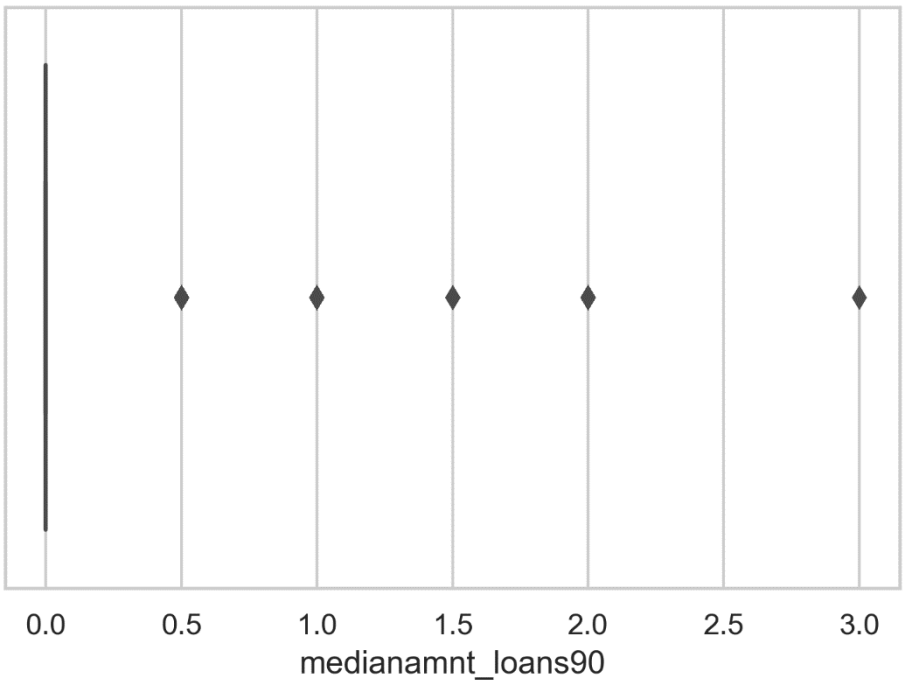

0
20000
40000
60000
80000
100000
maxamnt_loans30

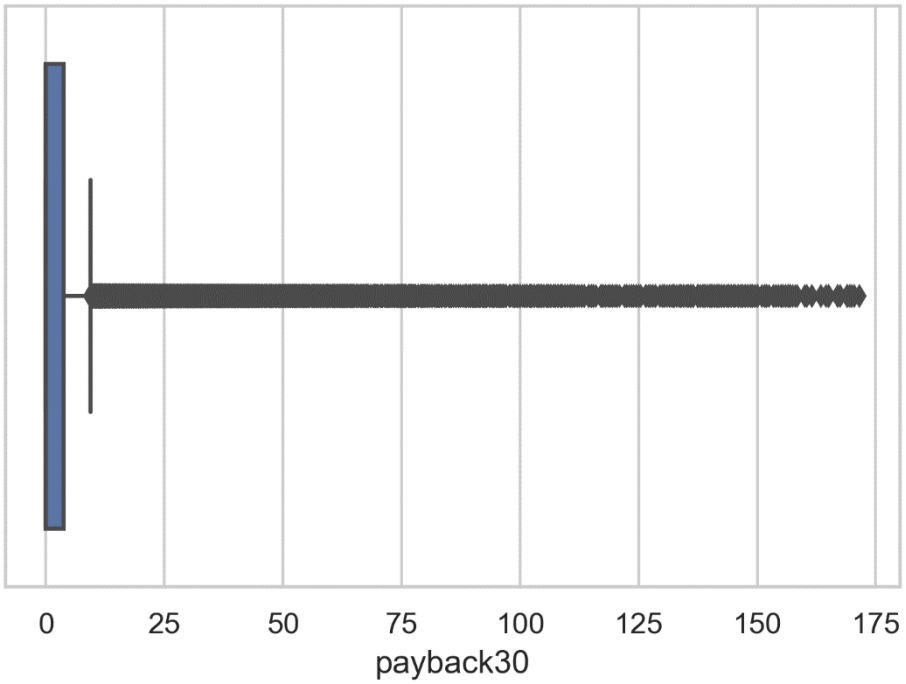

0.0
0.5
1.0
1.5
2.0
2.5
3.0
medianamnt_loans30

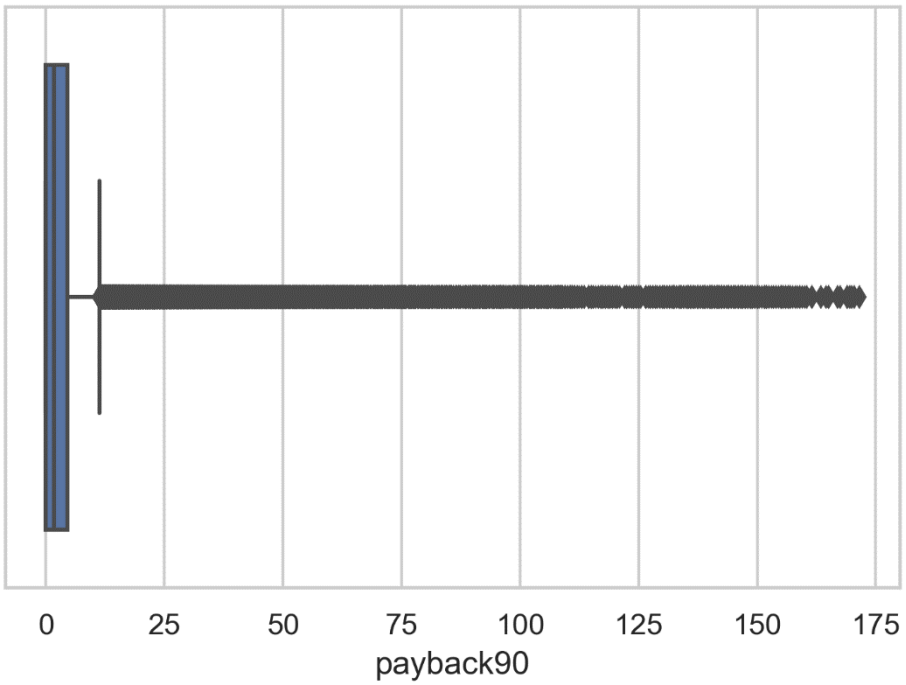

0
1000
2000
3000
4000
5000
cnt_loans90

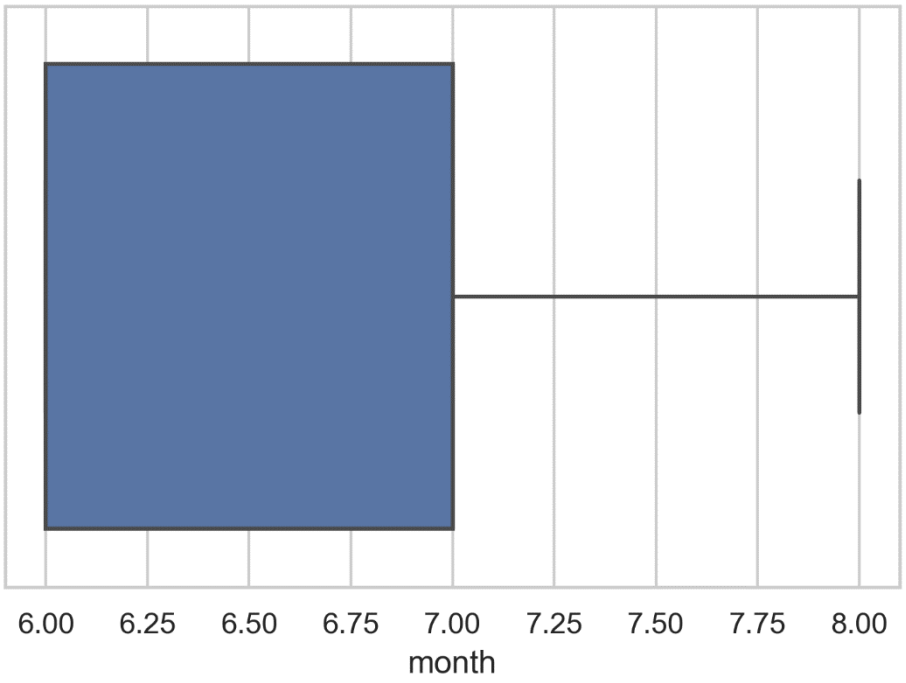

0
100
200
300
400
amnt_loans90

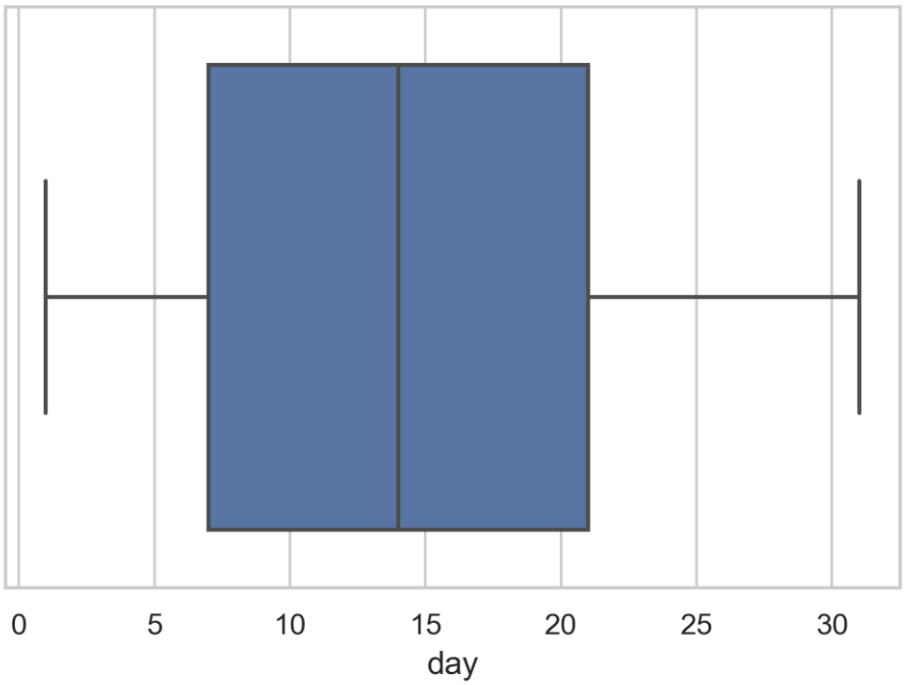

0
2
4
6
8
10
12
maxamnt_loans90

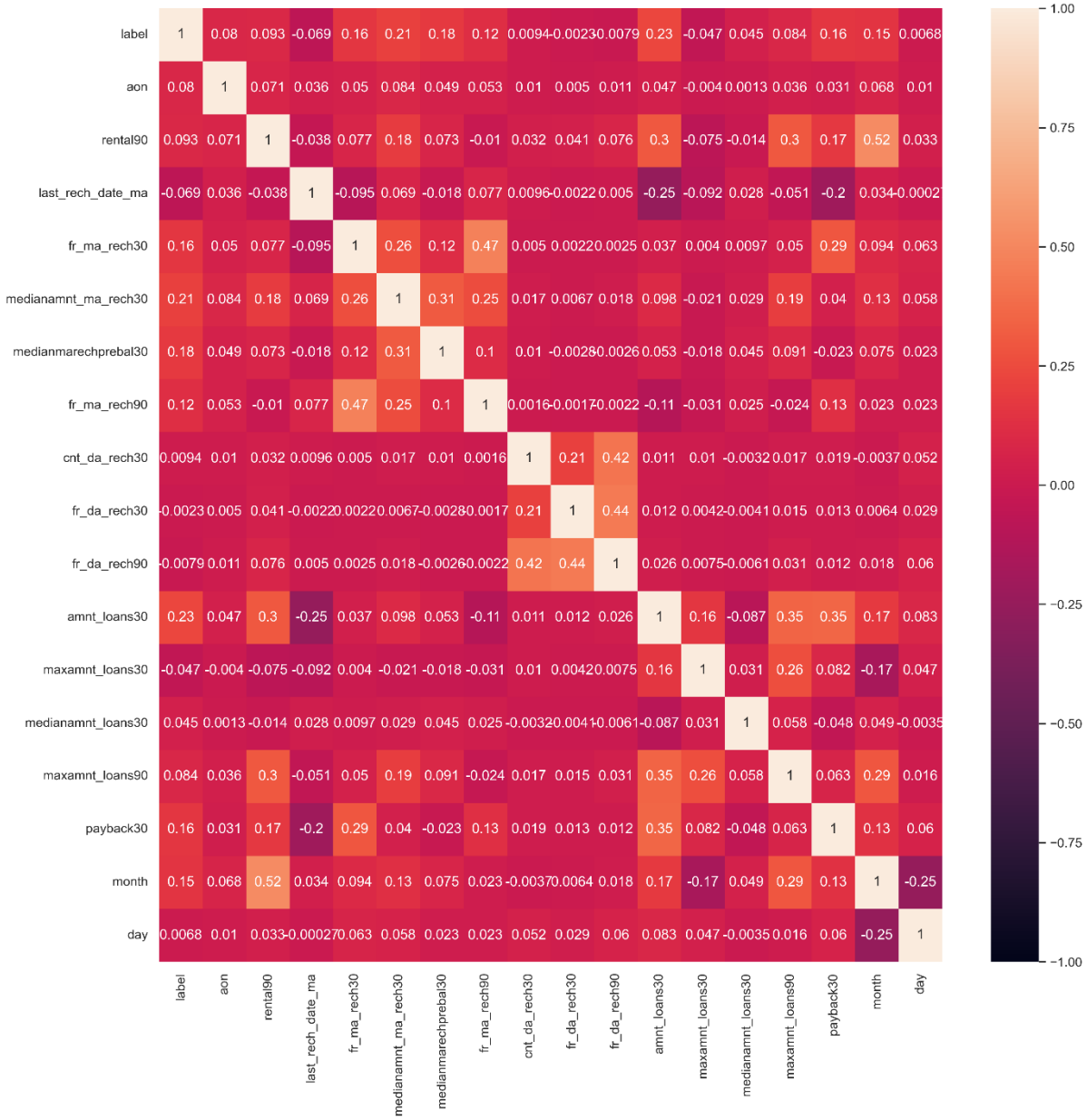

0.0
0.5
1.0
1.5
2.0
2.5
3.0
medianamnt_loans90

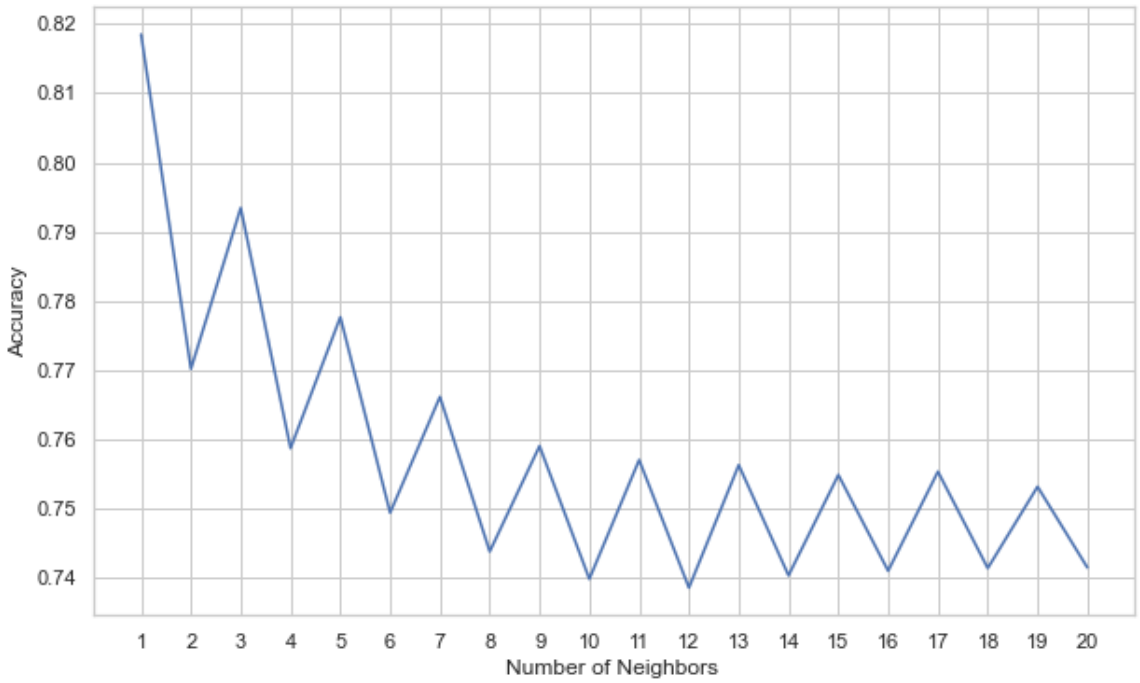

0
25
50
75
100
125
150
175
payback30

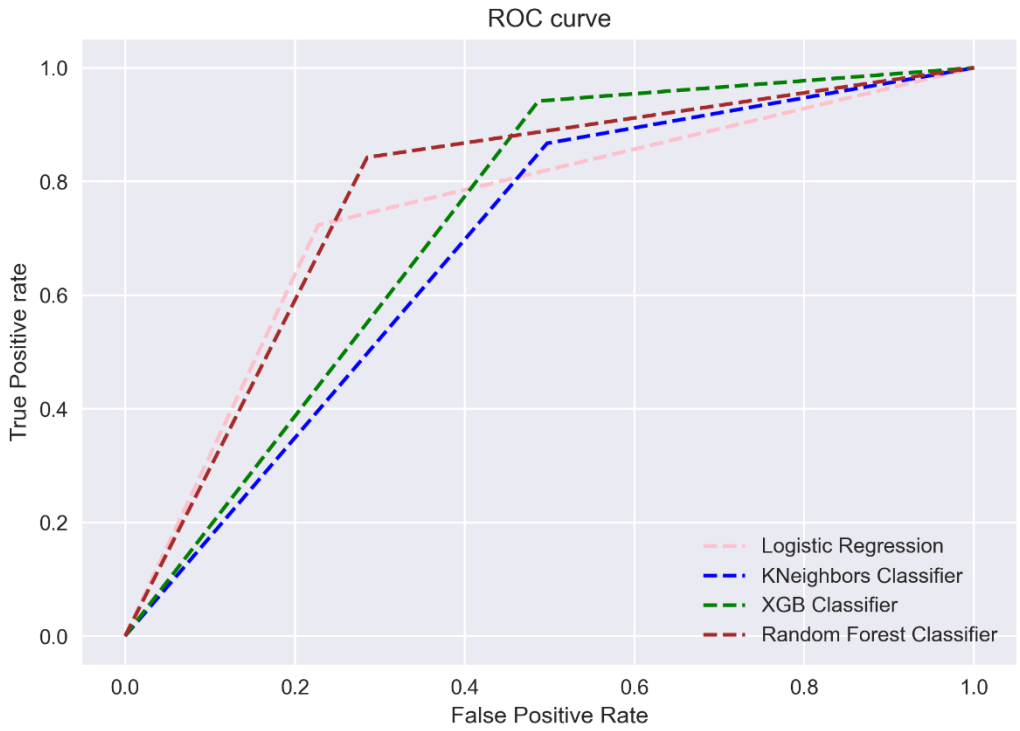

0
25
50
75
100
125
150
175
payback90


6.00
6.25
6.50
6.75
7.00
7.25
7.50
7.75
8.00
month


0
5
10
15
20
25
30
day


label 1 0.08 0.093 -0.069 0.16 0.21 0.18 0.12 0.0094 -0.0023 0.0079 0.23 -0.047 0.045 0.084 0.16 0.15 0.0068
aon 0.08 1 0.071 0.036 0.05 0.084 0.049 0.053 0.01 0.005 0.011 0.047 -0.004 0.0013 0.036 0.031 0.068 0.01
rental90 0.093 0.071 1 -0.038 0.077 0.18 0.073 -0.01 0.032 0.041 0.076 0.3 -0.075 -0.014 0.3 0.17 0.52 0.033
last_rech_date_ma -0.069 0.036 -0.038 1 -0.095 0.069 -0.018 0.077 0.0096 -0.0022 0.005 -0.25 -0.092 0.028 -0.051 -0.2 0.034 -0.0002
fr_ma_rech30 0.16 0.05 0.077 -0.095 1 0.26 0.12 0.47 0.005 0.0022 0.0025 0.037 0.004 0.0097 0.05 0.29 0.094 0.063
medianamnt_ma_rech30 0.21 0.084 0.18 0.069 0.26 1 0.31 0.25 0.017 0.0067 0.018 0.098 -0.021 0.029 0.19 0.04 0.13 0.058
medianmarechprebal30 0.18 0.049 0.073 -0.018 0.12 0.31 1 0.1 0.01 -0.0028 0.0026 0.053 -0.018 0.045 0.091 -0.023 0.075 0.023
fr_ma_rech90 0.12 0.053 -0.01 0.077 0.47 0.25 0.1 1 0.0016 -0.0017 0.0022 -0.11 -0.031 0.025 -0.024 0.13 0.023 0.023
cnt_da_rech30 0.0094 0.01 0.032 0.0096 0.005 0.017 0.01 0.0016 1 0.21 0.42 0.011 0.01 -0.0032 0.017 0.019 -0.0037 0.052
fr_da_rech30 -0.0023 0.005 0.041 -0.0022 0.0022 0.0067 -0.0028 0.0017 0.21 1 0.44 0.012 0.0042 -0.0041 0.015 0.013 0.0064 0.029
fr_da_rech90 -0.0079 0.011 0.076 0.005 0.0025 0.018 -0.0026 0.0022 0.42 0.44 1 0.026 0.0075 -0.0061 0.031 0.012 0.018 0.06
amnt_loans30 0.23 0.047 0.3 -0.25 0.037 0.098 0.053 -0.11 0.011 0.012 0.026 1 0.16 -0.087 0.35 0.35 0.17 0.083
maxamnt_loans30 -0.047 -0.004 -0.075 -0.092 0.004 -0.021 -0.018 -0.031 0.01 0.0042 0.0075 0.16 1 0.031 0.26 0.082 -0.17 0.047
medianamnt_loans30 0.045 0.0013 -0.014 0.028 0.0097 0.029 0.045 0.025 -0.0032 0.0041 -0.0061 -0.087 0.031 1 0.058 -0.048 0.049 -0.0035
maxamnt_loans90 0.084 0.036 0.3 -0.051 0.05 0.19 0.091 -0.024 0.017 0.015 0.031 0.35 0.26 0.058 1 0.063 0.29 0.016
payback30 0.16 0.031 0.17 -0.2 0.29 0.04 -0.023 0.13 0.019 0.013 0.012 0.35 0.082 -0.048 0.063 1 0.13 0.06
month 0.15 0.068 0.52 0.034 0.094 0.13 0.075 0.023 -0.0037 0.0064 0.018 0.17 -0.17 0.049 0.29 0.13 1 -0.25
day 0.0068 0.01 0.033 -0.0002 0.063 0.058 0.023 0.023 0.052 0.029 0.06 0.083 0.047 -0.0035 0.016 0.06 -0.25 1
label
aon
rental90
last_rech_date_ma
fr_ma_rech30
medianamnt_ma_rech30
medianmarechprebal30
fr_ma_rech90
cnt_da_rech30
fr_da_rech30
fr_da_rech90
amnt_loans30
maxamnt_loans30
medianamnt_loans30
maxamnt_loans90
payback30
month
day
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00





Observations:

1. age of cellular does not seem to have any effect on the label outcome as they are similarly spread for both the labels

2. subscribers with less daily amount spend are the majority ones who are failing to payback

3. subscribers with less average main balance are more unlikely pay the loan within the due date

4. Users who are recharging frequently are the once who are failing to pay back the loan taken.

5. Users who are recharging less than 100000 Indonesian Rupiah are mostly the user who are not paying back on time.

6. Users who are taking small loans amounts are failing payback the loan amount on time.

7. Users whose median of loan amounts is less than 1.5 are mostly probably the ones who are not paying the loan on time.

8. Users with zero loan amount have 100% success rate.

9. User of august month have 100% success rate.

10. Most of the data right hand skewed.

11. Majority of the columns have outliers in them.

Results:

Out of all the Classifier models Random Forest has performed well, XGB has some over fitting compared all the other model i.e., Random Forest have good precision and recall value compared to all the other models. Also, the predicted values and true value have good fit in case of Random Forest.

Conclusion:

In the whole dataset amount of loan and median of loan amount in last 30 days column has the most influence on the label columns. Payback30 and month columns are the other factors which have good influence on the label prediction.

Outcome of the Study:

Visualizing data helped to negotiate few outliers and biased data. Data Cleaning helps in minimizing the overfitting created during model training and improves the model performance. Random Forest can neglect outliers even when the data is fed with outliers to the machine learning model. XGB Classifier is not worth the use in this type of problem as it required ample amount of time. Simplifying the data was the most challenge part in this project but it overcome if minimal domain knowledge is exploited into this project.

Limitations of this work and Scope for Future Work:

The label predicted are only limited to particular cellular telecom industry. When we try to predict for different cellular network of another country the machine learning model will fail. To improve the model efficiency, we can add more of failure label in order to balance the data and get much improved precision, recall and f1-score for the given model.