Define task

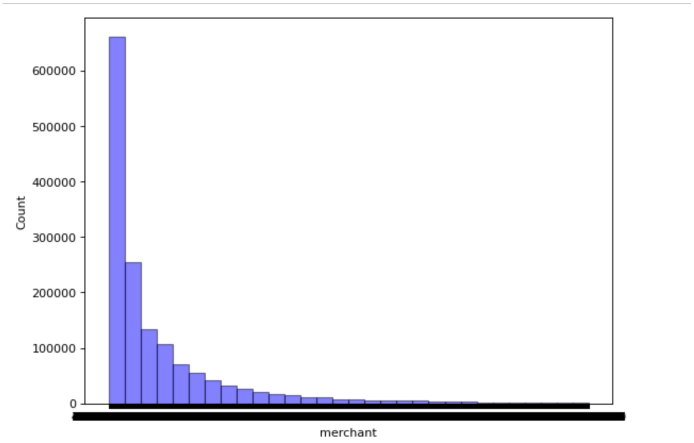
Sometimes a merchant may stop processing with Stripe, which we call churn. For this task, I am predicting the churning for all the given stripe merchants from the dataset, either will churn/have already churned or are not churning/have no potential to churn. The dataset contains 1513719 rows of data that each row corresponds to a transaction between 2033.1.1 to 2034.12.31 and there are a total of 14351 different merchants that occur transactions. There are three columns from the data: merchant name, exact date and time of the transaction. and amount of each transaction. From the given raw data, there is no label showing whether churn or not of each merchant. Besides, from the data, it's not easy to intuitively think of whether any particular merchant will face churning or not. Therefore, I am deciding to do some data preprocessing and data augmentation to better find out a criteria to define whether the merchant is a churn or not churn label. After that, I will apply some simple models to predict the churn. Besides predicting the churn, I will also analyze and understand the type of merchant processing with Stripe by looking at his total transaction count and amount, average transaction count and amount totally, monthly and quarterly and use one of the merchants for case study.

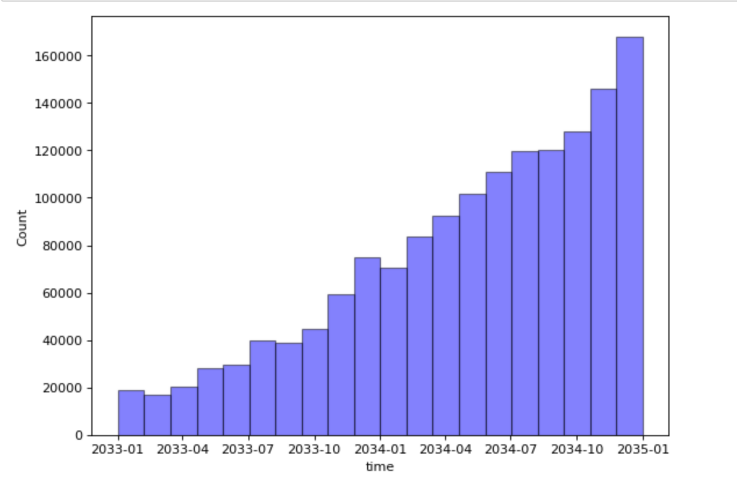
Assumption

Before beginning, it is important to make assumptions for the final result. Since there are all the transactions between merchants and Stripe, a highly potential company, and the data is in real life situation, I assume that there is a high non-churn and a low churn rate, or in other words it is unlikely for most of the merchants to stop processing with Stripe.

Data Preprocessing and Augmentation

First of all, I want to know the data type, structure as well as the distribution of merchants and time that occurs in transactions. I found out that there is no null or missing data, which is good. I also plot the histogram to better visualize the distribution.





I can see that the frequency of merchants is highly skewed to the right and frequency of time is almost linearly increasing throughout the time span. It indicates that with the time flies, more transactions occur. Although it is not good to have more skewness data, I decide to keep all the merchant data because it is likely that merchants with less frequency have only happened a few transactions, and if I remove those data, it may affect the diversity of data and affect the model prediction. It is good to see more transactions happen in the second half of time, because I am predicting the potential of churn/not churn at the end of 2034 so that the more transactions occur at the end, the better results I can get.

After understanding the basic structure and information of data, I decided to create a criteria to define churn/not churn. Intuitively, I can check the frequency distribution group by each merchant. That is, for each merchant, I will label ‘churn’ for those frequency that have overall decreasing trends, in other words those transactions occur more at the beginning of time span than occur at the end. It is time consuming and may cost overfitting if we compare each trade occurring for every merchant because there are a total of 1513719 trades. Therefore, I decide to group by month and compare the monthly frequency. Therefore, there are a maximum of 24 months and for some merchants there might occur trade for all 24 months.

Besides month, I also choose to group by seasonal quarter. There are 4 quarters per year, and there are 8 quarters in this dataset. For example, if merchant ‘A’ has a trade occurring in April 2034, then it will return April for the ‘month’ column and 6 for the ‘quarter’ column. I will also use the ‘Year_and_month’ column to differentiate the month in 2033 and in 2034, and it will return ‘2034-04’. For every merchant, I find out the retention rate, which

is the opposite of churn rate for monthly, quarterly and overall. Retention rate is calculated by the difference of frequency between the current section and previous section and then divided by the frequency of the previous section. A positive retention rate means that the merchant is retained and the higher the value, the higher rate of retention, whereas a negative rate means otherwise. If no trade happens before this section, it will return the result of 0. Note that I will ensure that I am calculating the retention rate from the previous section that occurs in any trade. For example, if there are trades occurring in quarter 1 and quarter 3 and no trade happened in quarter 2, then the quarterly retention rate is calculated based on quarter 3 and quarter 1.

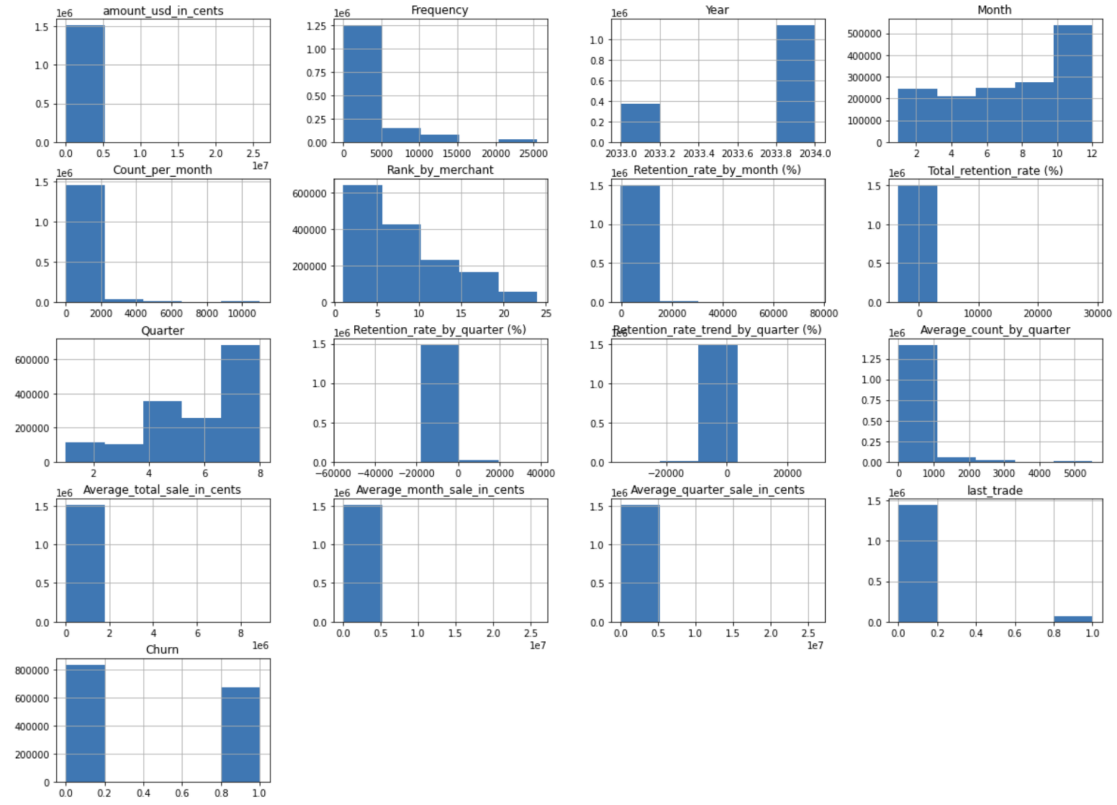
I also notice that in a real life case, there might be a situation where there is an increasing retention rate at the end but the overall retention rate is below 0. I am considering this type of situation as not churning, which makes more sense in a real life situation. The main reason behind this is that there is a sudden churning at the beginning of the time span but a retention signal at the end. Therefore, I decided to calculate the quarterly retention trend as the 'Retention_rate_trend_by_quarter' column based on the last three quarters that happened a trade. If there are not enough quarters, then simply calculate based on the entire quarters.

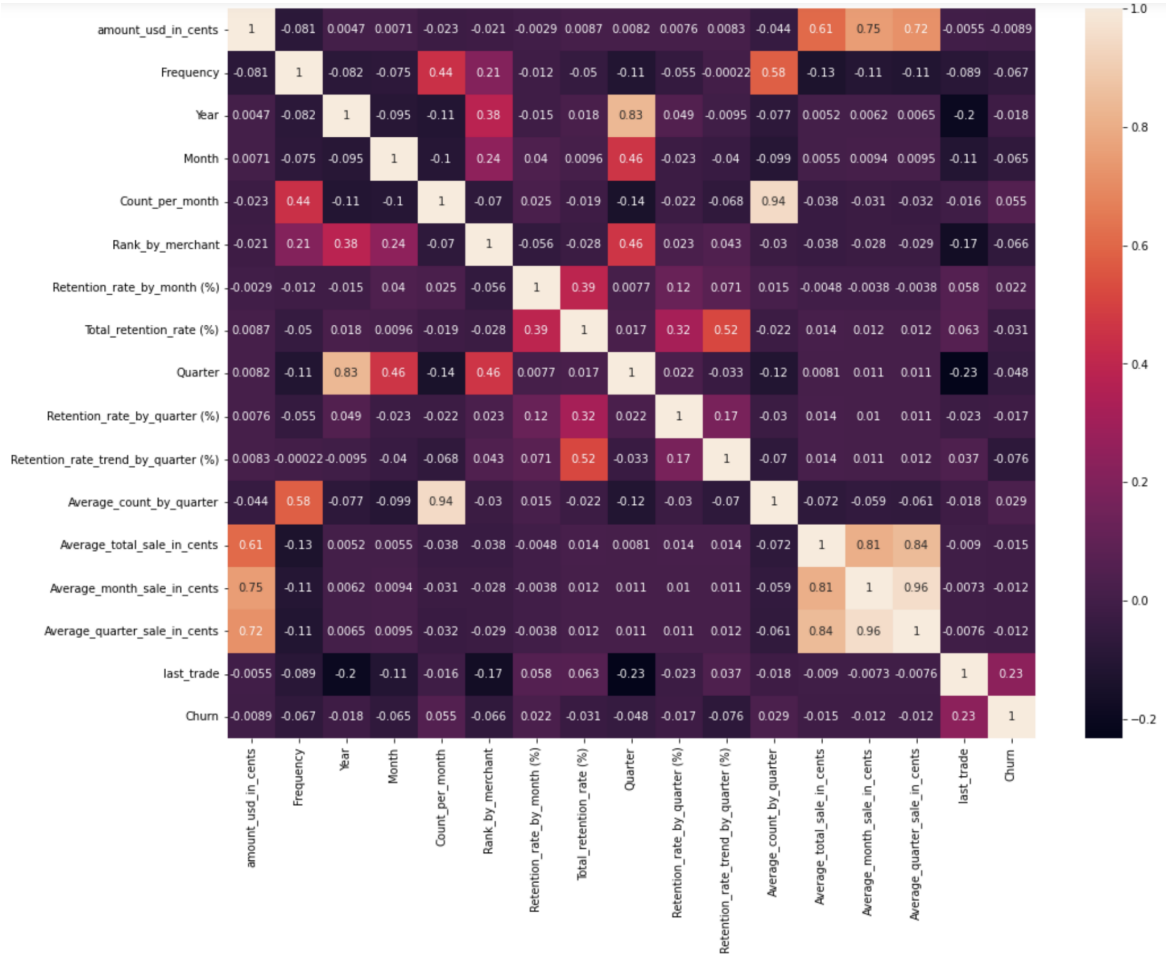
Lastly, I also check for the last time each merchant occurs a trade. If there are at least two quarters not occuring any trade, that is, if the last trade of a merchant occurs before 2034 July, and at the same time the entire trade count is larger than 100, then I should label it as 'Churn' because it has already churned out from Stripe. I am setting the threshold of 100 to avoid situations with merchants that have very few trades and we cannot determine whether churning by this method. I also augment some useful information such as the average trade frequency by each merchant quarterly, average sale by each merchant entirely, monthly and quarterly for later training and testing in models and analyzing the transaction activity.

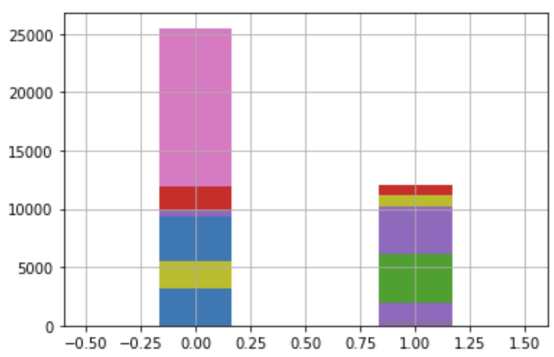
Define Churn

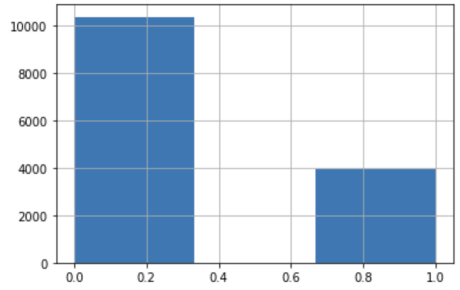
After finishing all the data preprocessing, I set the criteria for the target column 'Churn', whether the merchant will churn/have already churned or are not churning/have no potential to churn by the following:

1. If the merchant has no trade after 2034 July and the total trade count in two years is larger than 100, then it is 'churn'.
2. For the remaining merchants, if there is a negative overall retention rate and at the same time a negative retention rate trend for the last three quarters that happen a trade, then it is 'churn'.
3. For the remaining merchants, labeled as 'not churn'.
4. The final histogram distribution of all the columns, correlation of each column, as well as the 'churn' label before and after grouping by merchant name are shown below.


amount_usd_in_cents
Frequency
Year
Month
Count_per_month
Rank_by_merchant
Retention_rate_by_month (%)
Total_retention_rate (%)
Quarter
Retention_rate_by_quarter (%)
Retention_rate_trend_by_quarter (%)
Average_count_by_quarter
Average_total_sale_in_cents
Average_month_sale_in_cents
Average_quarter_sale_in_cents
last_trade
Churn


amount_usd_in_cents 1 -0.081 0.0047 0.0071 -0.023 -0.021 -0.0029 0.0087 0.0082 0.0076 0.0083 -0.044 0.61 0.75 0.72 -0.0055 -0.0089
Frequency -0.081 1 -0.082 -0.075 0.44 0.21 -0.012 -0.05 -0.11 -0.055 -0.00022 0.58 -0.13 -0.11 -0.11 -0.089 -0.067
Year 0.0047 -0.082 1 -0.095 -0.11 0.38 -0.015 0.018 0.83 0.049 -0.0095 -0.077 0.0052 0.0062 0.0065 -0.2 -0.018
Month 0.0071 -0.075 -0.095 1 -0.1 0.24 0.04 0.0096 0.46 -0.023 -0.04 -0.099 0.0055 0.0094 0.0095 -0.11 -0.065
Count_per_month -0.023 0.44 -0.11 -0.1 1 -0.07 0.025 -0.019 -0.14 -0.022 -0.068 0.94 -0.038 -0.031 -0.032 -0.016 0.055
Rank_by_merchant -0.021 0.21 0.38 0.24 -0.07 1 -0.056 -0.028 0.46 0.023 0.043 -0.03 -0.038 -0.028 -0.029 -0.17 -0.066
Retention_rate_by_month (%) -0.0029 -0.012 -0.015 0.04 0.025 -0.056 1 0.39 0.0077 0.12 0.071 0.015 -0.0048 -0.0038 -0.0038 0.058 0.022
Total_retention_rate (%) 0.0087 -0.05 0.018 0.0096 -0.019 -0.028 0.39 1 0.017 0.32 0.52 -0.022 0.014 0.012 0.012 0.063 -0.031
Quarter 0.0082 -0.11 0.83 0.46 -0.14 0.46 0.0077 0.017 1 0.022 -0.033 -0.12 0.0081 0.011 0.011 -0.23 -0.048
Retention_rate_by_quarter (%) 0.0076 -0.055 0.049 -0.023 -0.022 0.023 0.12 0.32 0.022 1 0.17 -0.03 0.014 0.01 0.011 -0.023 -0.017
Retention_rate_trend_by_quarter (%) 0.0083 -0.00022 -0.0095 -0.04 -0.068 0.043 0.071 0.52 -0.033 0.17 1 -0.07 0.014 0.011 0.012 0.037 -0.076
Average_count_by_quarter -0.044 0.58 -0.077 -0.099 0.94 -0.03 0.015 -0.022 -0.12 -0.03 -0.07 1 -0.072 -0.059 -0.061 -0.018 0.029
Average_total_sale_in_cents 0.61 -0.13 0.0052 0.0055 -0.038 -0.038 -0.0048 0.014 0.0081 0.014 0.014 -0.072 1 0.81 0.84 -0.009 -0.015
Average_month_sale_in_cents 0.75 -0.11 0.0062 0.0094 -0.031 -0.028 -0.0038 0.012 0.011 0.01 0.011 -0.059 0.81 1 0.96 -0.0073 -0.012
Average_quarter_sale_in_cents 0.72 -0.11 0.0065 0.0095 -0.032 -0.029 -0.0038 0.012 0.011 0.011 0.012 -0.061 0.84 0.96 1 -0.0076 -0.012
last_trade -0.0055 -0.089 -0.2 -0.11 -0.016 -0.17 0.058 0.063 -0.23 -0.023 0.037 -0.018 -0.009 -0.0073 -0.0076 1 0.23
Churn -0.0089 -0.067 -0.018 -0.065 0.055 -0.066 0.022 -0.031 -0.048 -0.017 -0.076 0.029 -0.015 -0.012 -0.012 0.23 1
amount_usd_in_cents
Frequency
Year
Month
Count_per_month
Rank_by_merchant
Retention_rate_by_month (%)
Total_retention_rate (%)
Quarter
Retention_rate_by_quarter (%)
Retention_rate_trend_by_quarter (%)
Average_count_by_quarter
Average_total_sale_in_cents
Average_month_sale_in_cents
Average_quarter_sale_in_cents
last_trade
Churn
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

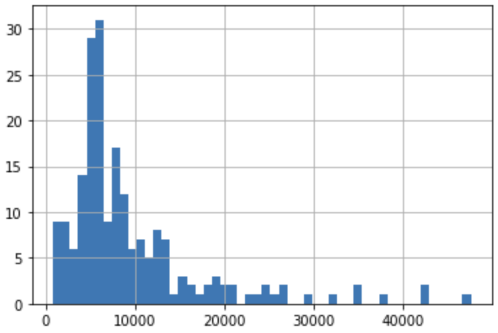





```
]:  0    10382
    1     3969
    Name: Churn, dtype: int64
```

There are a total of 14351 merchants and 10382 of them are not churning, whereas 3969 of them will churn or have already churned. **This is consistent with my assumption that there should be more merchants that are not churning.**

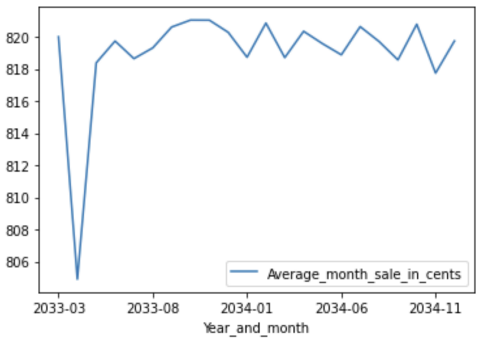
Merchant Transaction Analysis

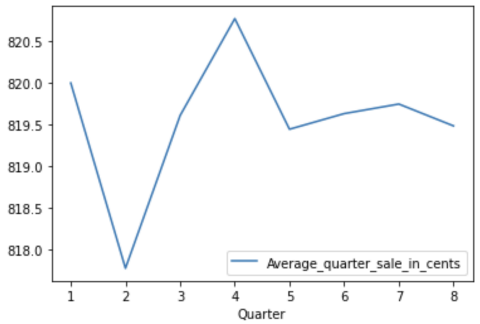



```
Out[93]: 0          819.675878
         25512      4987.168665
         37690      3441.583209
         49732      1508.438466
         61701     13466.075120
                        ...
         662619    29544.937102
         664034     5237.399576
         665448     4422.370746
         666829     5597.100366
         668194    19836.132062
         Name: Average_total_sale_in_cents, Length: 200, dtype: float64
```

In addition, I find and observe the average transaction amount of the top 200 merchants sorted by the highest transaction frequency. I can infer from the histogram that most merchants have the average transaction amount between 0-10000 in cents. For merchant ‘5608f200cf’, the one with the highest frequency (25512) of transactions with Stripe, I also visualize his average transaction amount monthly and quarterly. There is not much fluctuation of the trade amount, as they are all around 818-820 in cents monthly and quarterly. His average transaction is 819.68, which is also consistent according to the plots below. Therefore, based on the transaction amount and activity analysis, I can infer that

between 2033 to 2034, merchant ‘5608f200cf’ is trading with a transaction amount around 818-820 in cents most of the time (monthly, quarterly and overall).



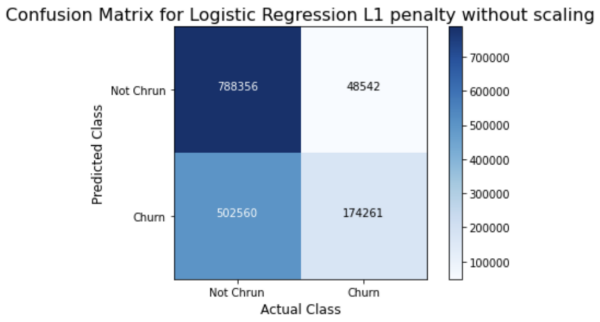


Model to Predict Churn

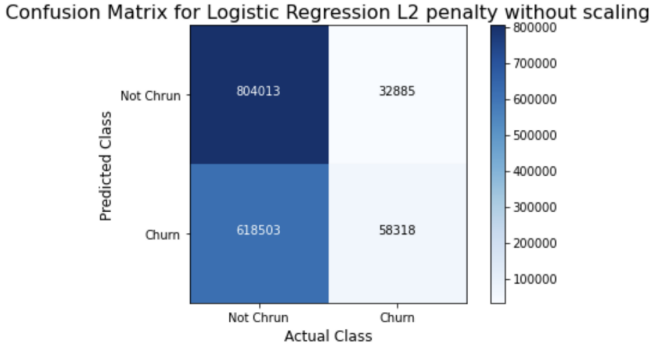
In the second half of the task, I use ‘Churn’ as the target to train and test datasets and then implemented classifiers using Logistic Regression(L1 and L2 penalty), and conduct k-fold cross validation to find out the best performing models, assess their cumulative performance across folds (report accuracy, precision, recall, and F1 score), and determine the best model for our particular data and visualize the confusion matrices. I also standardize training variables to check whether scaling brings any improvement to the model. Therefore, there are a total of 4 cases. The best model based on accuracy is 63.59% using Logistic Regression L1 penalty without scaling, which is only at the mediocre level . I only selected Linear regression as the model since it is simple. I believe that using Random Forest and Neural Network may improve the accuracy.

The following are the 4 cases results:

```
Cross Validation accuracy scores for Logistic Regression L1 penalty without scaling is: [0.63398779 0.63556668 0.63552044 0.63792511 0.63
511746 0.63705309
 0.63512407 0.63496552 0.63636604 0.63749992]
Cross Validation accuracy: 0.635912610984804 +/- 0.001191865735013186


accuracy:  0.6359284649264494
f1 score:  0.5642046877469232
precision score:  0.6964127217721746
recall score:  0.5997337793016009

              precision    recall  f1-score   support

           0       0.61      0.94      0.74    836898
           1       0.78      0.26      0.39    676821

    accuracy                           0.64   1513719
   macro avg       0.70      0.60      0.56   1513719
weighted avg       0.69      0.64      0.58   1513719

[[788356  48542]
 [502560 174261]]
```




```
Cross Validation accuracy scores for Logistic Regression L2 penalty without scaling is: [0.56767434 0.5697487  0.56355865 0.59881616 0.56
539519 0.56916074
 0.5712351  0.56417303 0.5625545  0.56445422]
Cross Validation accuracy: 0.5696770634293143 +/- 0.010098861068776608


accuracy:  0.569677066879652
f1 score:  0.43178275667516286
precision score:  0.602317805485247
recall score:  0.5234353338966361

              precision    recall  f1-score   support

           0       0.57      0.96      0.71    836898
           1       0.64      0.09      0.15    676821

    accuracy                           0.57   1513719
   macro avg       0.60      0.52      0.43   1513719
weighted avg       0.60      0.57      0.46   1513719

[[804013  32885]
 [618503  58318]]
```

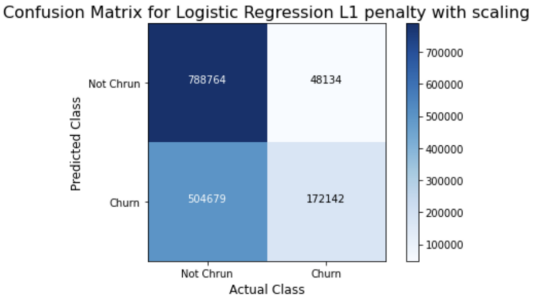


```
Cross Validation accuracy scores for Logistic Regression L1 penalty with scaling is: [0.63239569 0.63411331 0.63447665 0.63726449 0.6332
432 0.63615464
 0.6334725  0.63320826 0.63470127 0.63660807]
Cross Validation accuracy: 0.6345669189863838 +/- 0.0015370346692724305
```

```
accuracy:  0.6347981362458951
f1 score:  0.5621402956869114
precision score:  0.695650296777035
recall score:  0.5984121300946266
```

	precision	recall	f1-score	support
0	0.61	0.94	0.74	836898
1	0.78	0.25	0.38	676821
accuracy			0.63	1513719
macro avg	0.70	0.60	0.56	1513719
weighted avg	0.69	0.63	0.58	1513719

```
[[788764  48134]
 [504679 172142]]
```




```
Cross Validation accuracy scores for Logistic Regression L2 penalty with scaling is: [0.63193325 0.63381603 0.63341305 0.63577808 0.63309
595 0.63598948
 0.63330735 0.63288455 0.63429829 0.63592102]
Cross Validation accuracy: 0.6340437041995978 +/- 0.0013442808168294745
```

```
accuracy:  0.6340437029594
f1 score:  0.5607327560101308
precision score:  0.6951895076147252
recall score:  0.5975239695804585
```

	precision	recall	f1-score	support
0	0.61	0.94	0.74	836898
1	0.78	0.25	0.38	676821
accuracy			0.63	1513719
macro avg	0.70	0.60	0.56	1513719
weighted avg	0.69	0.63	0.58	1513719

```
[[789079  47819]
 [506136 170685]]
```