Stripe Merchants Clustering and Churn Predictions

Zhaox334@umn.edu

Introduction

In this project, a two-year transaction history data from Stripe's merchant was given to achieve two goals. The first is merchant segmentation and the second is customer churn prediction.

For question 1, I created features for individual merchants and trained two clustering models with unsupervised learning algorithms after deciding the best number of clusters and hyper-parameter tuning. I was able to visualize my clustering result and have an interpretation of individual clusters from the center of the clusters.

For question 2, I first defined churn and created churn labels for the merchants in both empirical and theoretical ways. I then compared, trained and hyper-tuned a classification model with the label and feature from part 1 and other time series features from the original data to predict future churns of individual merchants. I tried to determine the reasons for the churn and provided some insights on how to proceed with the churning merchants.

The report will describe my approach to solving the two questions. Starting with my understanding of the data and feature engineering, followed with my choice of models, hyper-parameters. And lastly some ideas on how to approach churn merchants.

Keywords: customer segmentation, unsupervised learning, churn prediction

1. Presumptions

There are some assumptions I made when I analyzed the dataset and trained the models, which could cause huge difference in data interpretation and model performance. It is important to state them in the first place.

- Active merchants mean they have transactions at the end of 2034.
- A churn means the time since last transaction is so long that it has less 10% chance of happening.
- If one merchant does not have transaction at the last day of 2034, I will not automatically take them as stop processing with Stripe and they are not taken as churn. My definition of churn is defined in section 4. This could cause a big difference in terms of model performance. One would have far more positive (churn) labels than another, hence, change the underlying data distribution.
- If merchants churn, they will not be back. Whether with the same id or a different id.
- Different merchant id means different merchant, there will not be two ids associated with same store or merchant.

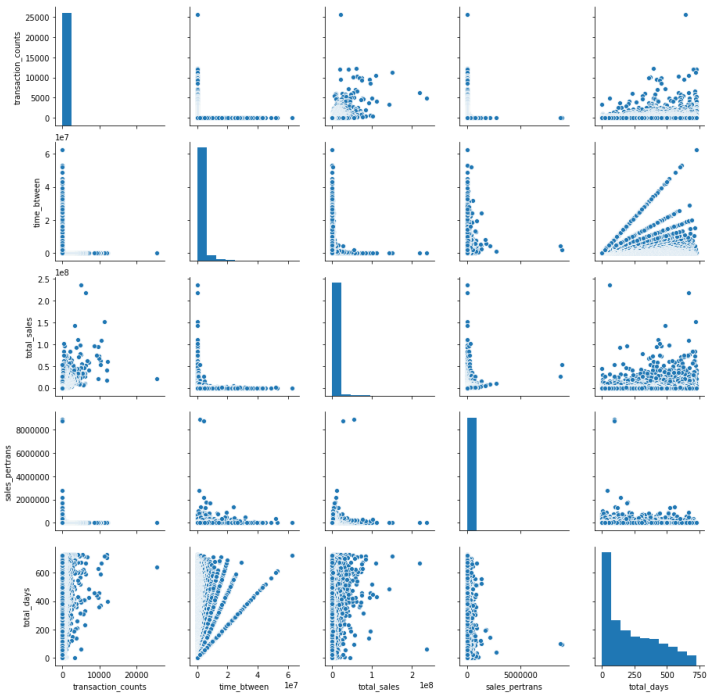
2. Exploratory Data Analysis and Feature Engineering and Data Preparation

This section introduces my understanding of the dataset and how I create new features for later problems.

2.1 EDA

The data has 1 million rows and 3 columns, and it is a time series of individual merchant's transaction with 2-year range. Data is clean but needs more features in order to apply machine learning models.

Since we are focusing on clustering and churn prediction instead of predicting future sales, my focus will be on merchant level features, which are retrieved from time series data.



Pair plot of features

2.2 Feature Engineering

The new features come with the following categories.

Transaction volume, transaction total amount, frequency, tenor with Strip and different time of the day (morning, night etc.).

- sales volume, amounts (mean, variance).
- Frequency of the transitions (daily, weekly etc.)
- when did sales happen (sales per day/week/month, time of the day morning, noon, night.
- start and date on strip, how long they have been as a customer.

Then, there is a divide between features to get per day features.

This is based on my experience working with retail time series. The idea is to use existing time series to come up with content or feature represent to separate merchants from different clusters.

I calculated VIF score to detect multicollinearity. There is only one feature that has about 30 VIF score. Since VIF does not necessarily affect model performance, unless they are perfectly correlated, I decided to keep all of them.

After working with the data more, I came up with more detailed information like below.

- booming business more likely to stay, what is the metric to identify them? trend? How to identify stable business, unhealthy business.
- instead of using mean, take the most recent average, with weights.
- small business to large business in long term, think about long term change underlying model distribution.

To find that information, I used TSFresh module to add additional features. TSFresh uses p value to identify important features from time series. Example would be trend, means, variance and categorical data like has_duplicate or not. It is massive and end up with 700+ features.

When dealing with supervised learning, the features could be condensed or purified to more important features. In my application here, I tried to use PCA to reduce dimension and integrate later models. The result was not as impressive as I thought, presumably due to the existing features catch good portion of the clustering problem. I can see it works with other harder prediction problems.

Question 1

3. Clustering and model training

This part talks about unsupervised learning to find out segment different merchants.

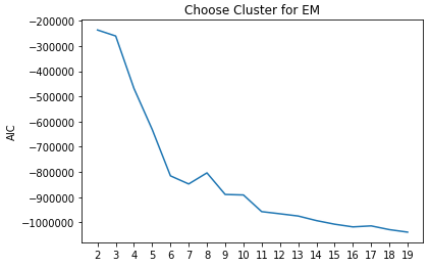
3.1 Metric consideration

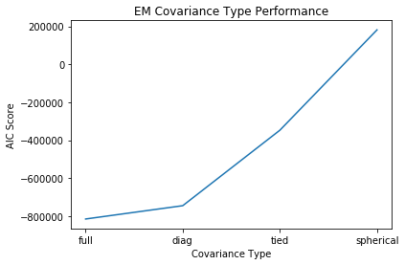
One important consideration when working on machine learning models is to use correct metric to measure success of different models. This could be tricky for unsupervised learning, as there are no labels to measure with. In this project, I chose three different metrics, AIC score, Silhouette Score and Inertia. They are used to describe how well your clusters have been separated.

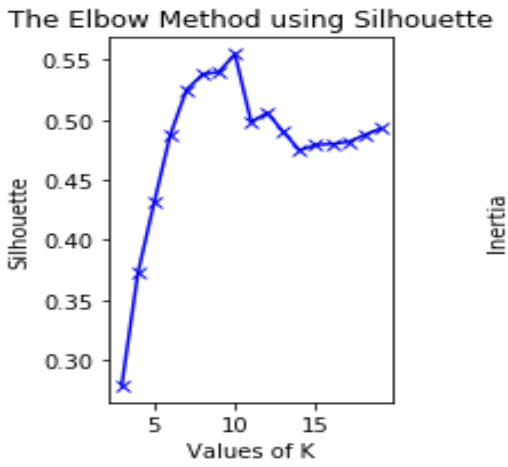
3.2 Model comparisons

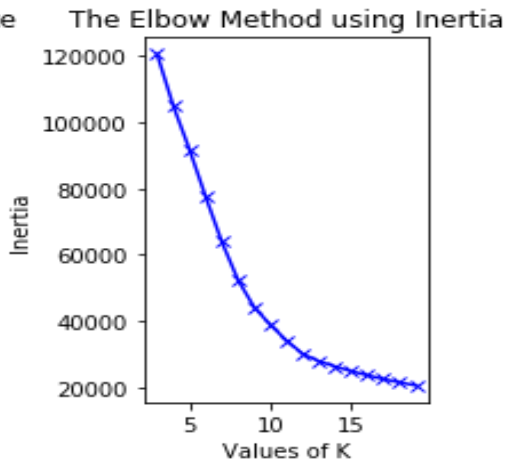
I chose KMeans Model and Gaussian mixture to identify the clusters. KMeans us distance to separate the datapoint while GM use MLE to estimate a distribution that fits the data the best.

One important hyper-parameter is the number of clusters. I used K-elbow plot and Silhouette plot to visualize the result.

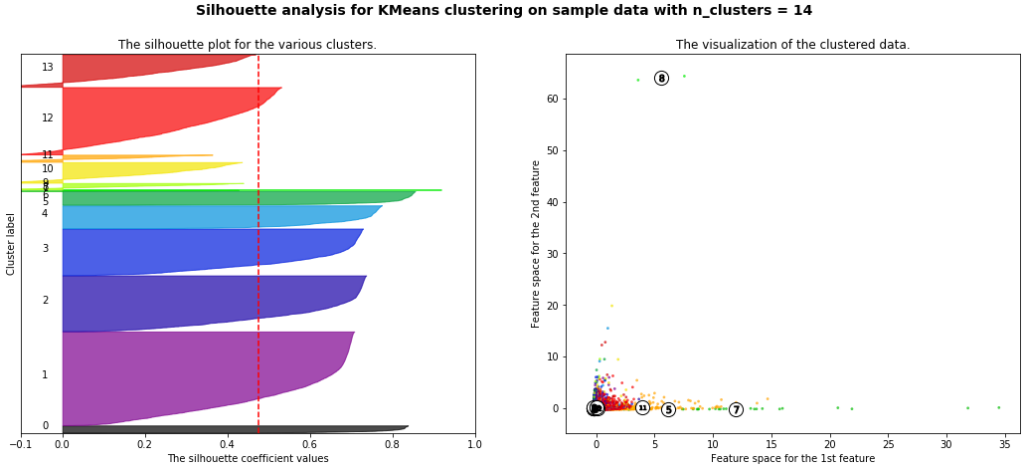








Number of clusters	Silhouette_score
4	0.373
8	0.538
10	**0.555**
14	0.475



3.3 Result Analysis

With number of cluster as 10 and other hyper-parameters and standardized input data, I trained the two models to get the labels. 90% of the labels from two models are the same (details in notebook). I implemented a model ensemble method in the next section. In this section, KMeans labels are used for interpretation.

After getting the centers, I inversed the center to find original data. For the 10 clusters, my understanding is as follows.

Clusters	Evidence	Hypothesis	My Estimates	Suggestion
2, 9	large amounts per transaction, large number of transactions	data error, internal test merchant	outliers	Look into data quality
3, 4, 5	transactions at the night, newer merchants, lower to medium transaction and amount	on board when stripe expend to international business in the late stage	international business	support in the night to help growth; language barrier
1	have been with stripe the longest, loyal customer, more transactions as well.	they trust stripe and use frequently	power users	use resource to guarantee power users retention
6	largest total volume, more transactions at noon, transaction is the third highest	people buying lunch during daytime	lunch dining service	Offer customized service like convenient way for tipping
7	lowest in transactions, sales per day, second longest history	small business could struggle with scaling up	Small business owners that has been with stripe since beginning	Offer service like SaaS CRM tools to scale up
0, 8	4^{th} and 5^{th} largest in terms of total transactions, transactions in the morning	shorter history with higher volume could mean they are growing	newer business that just starts to use Strip	provide help for better growth for Stripe to grow exponentially

	transaction_counts	time_btween	total_sales	sales_pertrans	total_days	trans_perday	sales_perday	time_of_day_Late Night	time_of_day_Early Morning
0	75.648377	1.528726e+06	1.162637e+06	3.428141e+04	218.615256	0.777713	1.160831e+04	-2.289835e-16	2.220446e-16
1	4228.265487	2.529346e+04	4.954747e+07	3.155151e+04	529.097345	9.550135	1.392021e+05	-3.816392e-17	3.539823e-02
2	3358.000000	1.580280e+01	4.384030e+07	1.305548e+04	1.000000	3358.000000	4.384030e+07	0.000000e+00	1.000000e+00
3	26.685714	2.741915e+06	3.262650e+05	2.336264e+04	80.195918	6.617550	8.338482e+04	1.000000e+00	5.551115e-17
4	10.824295	2.222333e+06	2.622124e+05	2.988760e+04	98.900217	1.017067	2.183794e+04	5.551115e-17	1.387779e-16
5	60.529489	1.812915e+06	1.171954e+06	3.742833e+04	178.966361	1.380754	2.349423e+04	-8.673617e-17	2.220446e-16
6	110.777638	1.227745e+06	1.923218e+06	3.115626e+04	266.568620	0.898319	1.437869e+04	-3.747003e-16	-2.164935e-15
7	2.851145	1.946777e+07	1.649370e+05	5.853011e+04	373.000000	0.008216	4.730528e+02	3.816794e-03	5.343511e-02
8	51.015424	1.749691e+06	7.011316e+05	3.120986e+04	142.964010	2.298618	3.631695e+04	5.898060e-17	1.000000e+00
9	4.500000	2.934809e+06	3.983537e+07	8.834725e+06	97.500000	0.046234	4.092882e+05	0.000000e+00	0.000000e+00

3.4 Clustering ensemble

Model ensemble is a way to combine different model results to achieve better result. The package I chose is ClusterEnsembles. The performance was not as what I expected. This would be a nice area for future work to first try different clustering models and then explore options for ensembles.

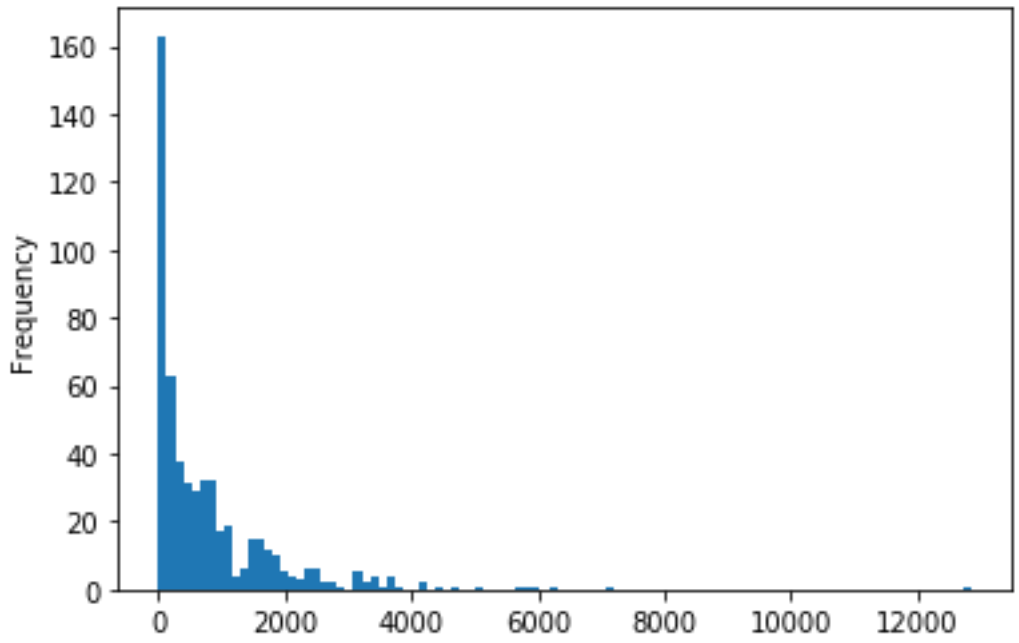
Question 2

4. Churn definition and assumption

I used time since the last transaction to identify churn. I used both empirical and theoretical ways to find the threshold for a churn.

4.1 Theoretical way with Exponential Distribution

Using historical data, the time between transactions is following an exponential distribution. I created 10 different distributions for 10 different clusters by calculating the rate and use the inverse CDF to get the threshold. Currently it is set as 90%. This is another big assumption that needs to be consulted with domain experts or doing experiments on.



Currently churn is assumed as 90%. This will need to be discussed and consulted with domain experts.

This also affects model performance. A less strict threshold will have a more positive label and could increase the complexity or change the distribution of the positive data, thus causing false negative or false positive.

4.2 Empirical way

The empirical way is just ranking the time in an ascending order and choosing the 90^{th} one as the threshold. It is easy to understand and faster to calculate.

Since there is no label to compare with, there will not be a way to know which performs better. Latter implementation is based on theoretical exponential model.

5. Churn prediction

We finally have labels! Since I chose 90% as the threshold for churn, the label is imbalanced with ratio about 1:10. I have used auc-roc score to balance between TPR and FPR.

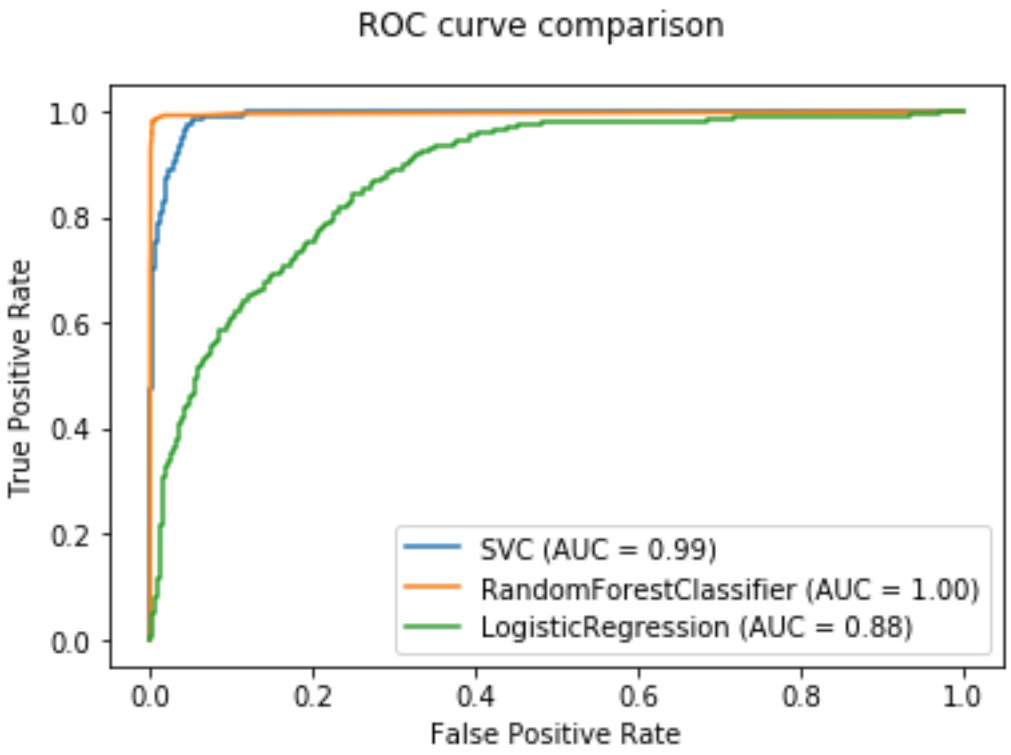
Before working on the models, I standardized the features for future use with scale sensitive models like SVM. I have also split the training and testing data.

To work with supervise learning, I usually use cross_val_score from sklearn and try different model with default parameters to get an understanding of which model works better for the problem. Below are my results.

It is not surprising that non-linear models work better than logistic regression. I chose Random Forest for better interpretation. I have also plotted ROC curve to find the best models.

5.1 Model comparison

RandomForestClassifier	GradientBoostingClassifier	LogisticRegression	DecisionTreeClassifier	SVC
0.996452	0.998817	0.891605	0.99862	0.969748



5.2 Model Result

After deciding the model, I used grid search to fine tune the hyper-parameters. And use the best parameter combo to train the model. I was able to achieve 0.996 accuracy. ROC also confirms the performance of the model.

5.3 Result Analysis

With the random forest model, I can find the most important feature.

	Importance
time_btween	0.391168
trans_perday	0.315995
cluster	0.061435
sales_perday	0.060556
transaction_counts	0.045646

The frequency and the trade volume rank high. Transforming that into business insight is to focus on getting the merchant to use Stripe more frequently, not necessarily dealing with large transaction amounts. Stripe could provide some incentive for merchants to use the service more often to avoid churn.

5.4 Find the active churn and work on it.

Out of 12k merchants, 601 of them have transactions at the end of 2034. I define them as active users.

I calculated the probability of churn with predict_prob method. The top likely churn merchants are selected for further analysis.

Out of 601, 9 merchants are churning according to my model. Again, this is under the assumption that the definition of churn is less than 10% chance of happening.

Many of them show proof of churning, for example decreasing in sales amount, less frequent transactions etc.

Clustering of the 9 is as follows.

Clusters	Counts
0	1
1	4
6	2
8	2

5.5 Suggestion for the business

If my hypothesis is correct on the clusters, business should pay immediate attention on cluster 1 as power user is important for revenue and short-term company goals.

For cluster 0 and 8, try to reach out to them and see if they have a problem learning to use Stripe. Thye individual maybe smaller in terms of transaction amount, but the number of the smaller business could be exponentially expanded and getting to know their need should be beneficial in the long run.

For cluster 6, maybe reach out to see if COVID situation affects downtown restaurant business. Workers are working from home and demand is just not as much as before. With that case, maybe offer some discount for the moment in terms of fees and connect them with government or institutional help.

It’s all about a balance between business cost and lifetime value. Short-term goal vs long term goal. Identifying the churn can help business to fast and accurately control the problem and clustering helps to identify the reason faster.

Summary

In this project, I created new features from historical merchant transaction data and use them trained unsupervised learning for segmentation and defined and identity churn both empirically and theoretically. Then used the label to train a classification model to predict active merchants' probability to churn.

There are a few parts I would like to explore further in the future. First is looking into extruded_features to hand-pick some valuable feature to help is training.

Secondly when dealing with dimension reduction, ICA, random projection are all valid approaches to try and compare with PCA.

Lastly, clustering ensemble is an active area where new ideas and packages are created. It would be nice to improve model performance with ensemble.

Reference

1. Christ, Maximilian, Andreas W. Kempa-Liehr, and Michael Feindt. "Distributed and parallel time series feature extraction for industrial big data applications." *arXiv preprint arXiv:1610.07717* (2016).

2. Vega-Pons, Sandro, and José Ruiz-Shulcloper. "A survey of clustering ensemble algorithms." *International Journal of Pattern Recognition and Artificial Intelligence* 25.03 (2011): 337-372.

3. On the Power of Ensemble: Supervised and Unsupervised Methods Reconciled* SDM'2010 Columbus, OH

4. Clustering Ensemble https://cse.buffalo.edu/~jing/cse601/fa12/materials/clustering_ensemble.pdf

5. ClusterEnsemble Pypi page https://pypi.org/project/ClusterEnsembles/

6. TSFresh https://tsfresh.readthedocs.io/en/latest/