B2B Customer Segmentation and Churn Prediction

Alon Benhaim

Business Challenge:

Given random sample of future merchant transaction activity (date, time, and amount), the goal is to try and answer the following questions:

1. Infer the types of merchants using the platform: identifying different kinds of businesses and generate assignments for each of them.
2. A merchant that stops processing with the platform is called “churner”. Define churner merchants, identify churners in the dataset and then build a model to predict which active merchants are most likely to churn in the future.

Translating those questions into data science language our goals are:

1. Merchant (in this case, B2B) segmentation.
2. Merchant churn prediction.

We will accomplish the first goal using unsupervised learning algorithm, and for the second we will build a supervised learning model.

Merchant segmentation insights are crucial for efficient allocation of marketing resources.

Dataset:

	Unnamed: 0	merchant	time	amount_usd_in_cents
0	1	faa029c6b0	2034-06-17 23:34:14	6349
1	2	ed7a7d91aa	2034-12-27 00:40:38	3854
2	3	5608f200cf	2034-04-30 01:29:42	789
3	4	15b1a0d61e	2034-09-16 01:06:23	4452
4	5	4770051790	2034-07-22 16:21:42	20203
...	...	...	...	...
1513714	1513715	72d37bedbf	2034-06-21 13:47:51	5274
1513715	1513716	5608f200cf	2034-04-20 02:23:59	754
1513716	1513717	fcbd1dae68	2033-09-19 14:02:33	13203
1513717	1513718	9843e52410	2034-12-28 20:07:59	4845
1513718	1513719	32acddd6cc	2034-08-23 09:07:07	3862

- The data consists of ~1.5M transactions of 14,351 different merchants over 2 years (from 01/01/2033 to 12/01/2034).
- Number of transactions per business range from 1 up to 25K.
- There are no refunds, no missing data values and all transactions are within the 2 yeas range.

Merchant Segmentation

Method and Assumptions:

Since we have a large number of businesses with a moderately large number of transactions overall, we won't study each business individually. Moreover, since we are dealing with a payment platform, and together with the second goal of churn prediction, we assume that we want to learn something about the business type so that we can keep businesses using the platform, attract new customers etc.

Based on past coursework experience and short literature survey we chose to do **RFM analysis** (Recency, Frequency, Monetary). RFM analysis gives you a base model to analyze merchants according to their transactional behavior - how recent was their last transaction, how often they purchase and how much are they spending and therefore is an effective method in our case.

We will segment the merchants using the **k-means clustering algorithm.** One of the major advantages of K-Means is that it can handle larger data sets like in our case.

The assumptions we make are:

- The dataset is a random sample of future merchant using the platform (it is not skewed towards any specific business in particular).
- Merchants who have processed with the platform recently are more likely to process again in the near future.
- Frequent merchants are more likely to process again than less frequent merchants.
- Merchants with a high total are more likely to have high total in the future than low total merchants.

Data Preprocessing:

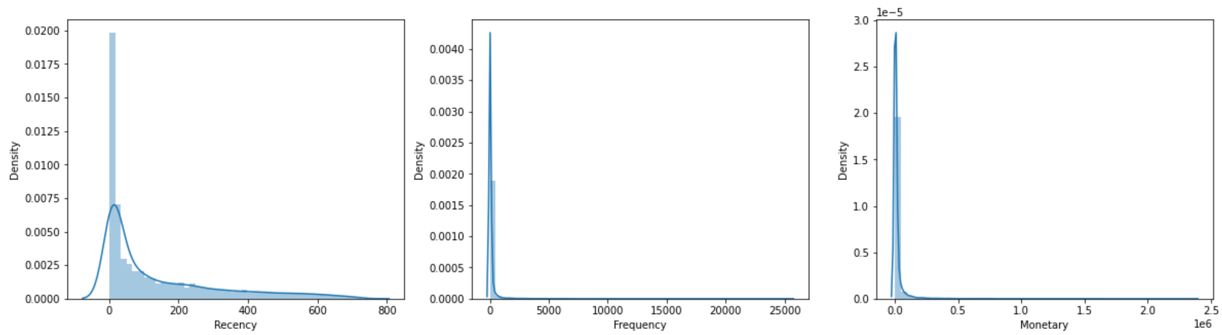
To this end, we aggregate the data to find the Recency, Frequency and Monetary of each business. Definitions:

- Recency (*0<Recency<730*) – The number of days between the data end date (12/31/2034) and the most recent merchant transaction date.
- Frequency – The total number of transactions.
- Monetary (in USD) – The total amount of dollars spent in all the merchant's transactions.

merchant	Recency	Frequency	Monetary
0002b63b92	595	1	33.79
0002d07bba	17	4	892.78
00057d4302	515	28	295.21
000bcff341	510	1	78.26
000ddbf0ca	578	1	102.99

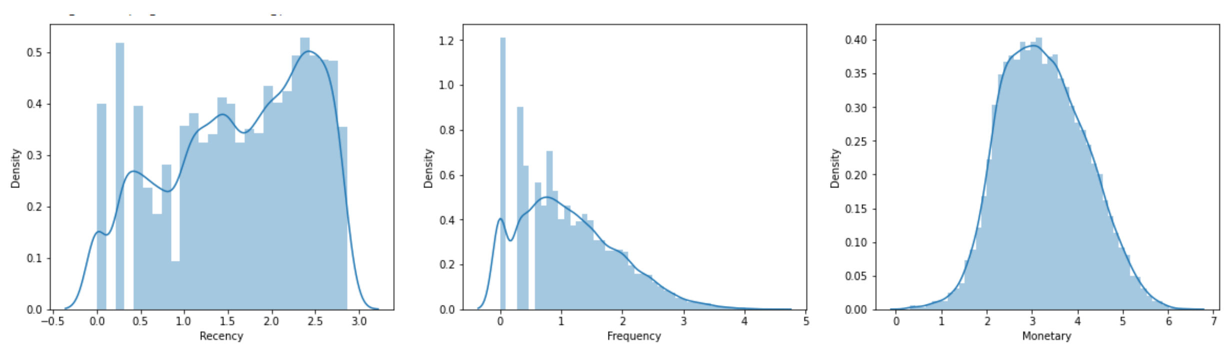
Note that we completely ignore the time of day for each transaction as our method selection and assumptions suggest we only care about the date and frequency.

Histogram plots:

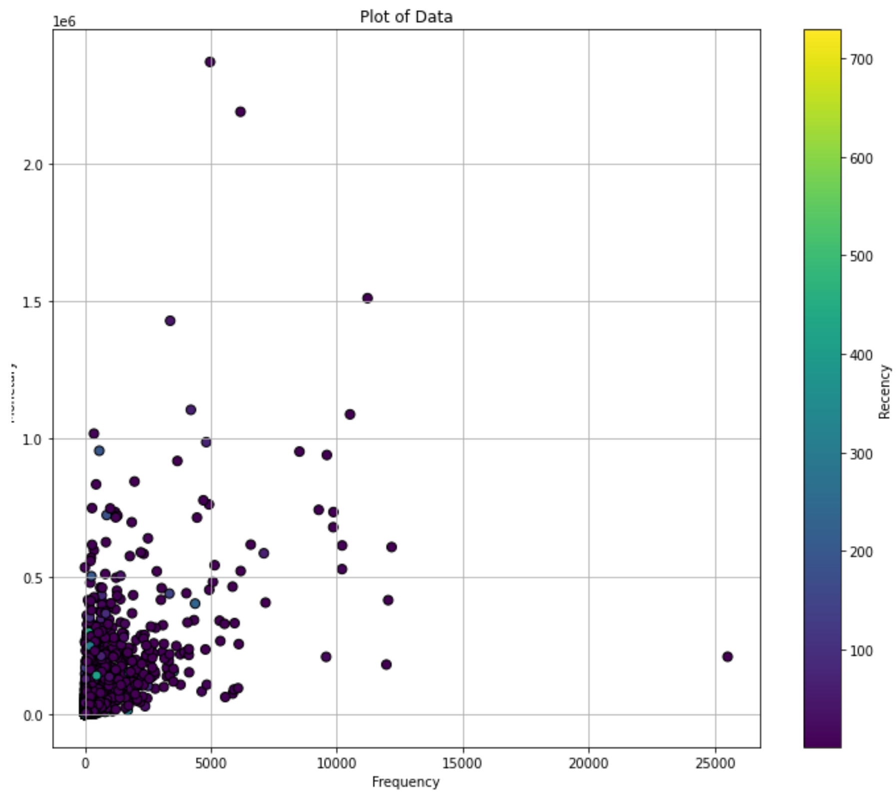


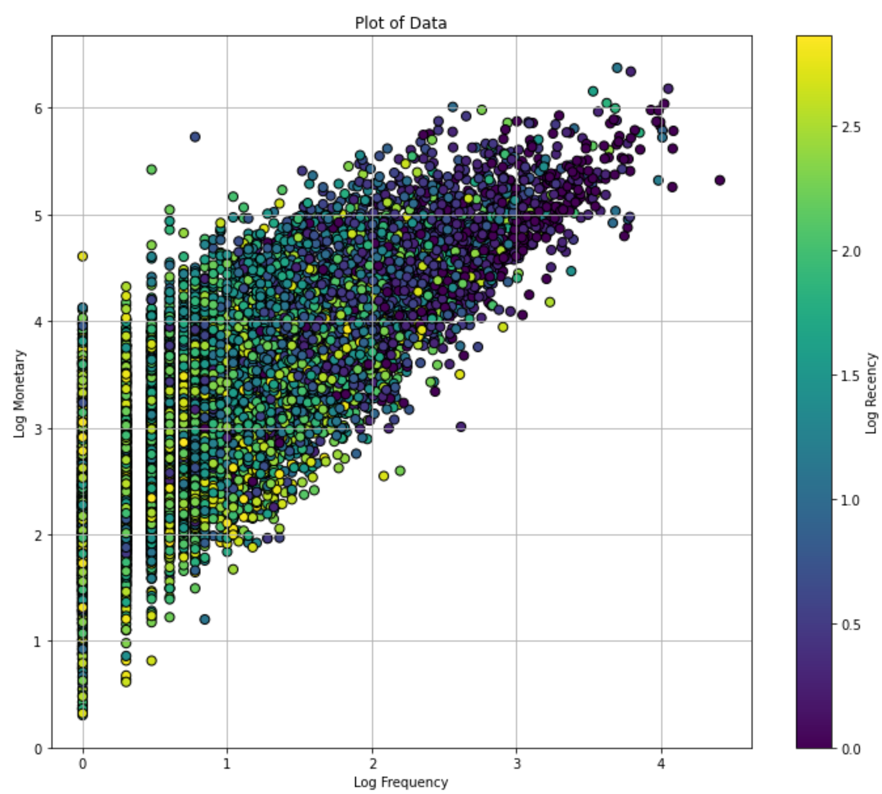
For clustering we need the data to have the same scale. Our features seem to conform to a power law distribution that clumps data at the low end.

Here are the histogram plots after log transformation:

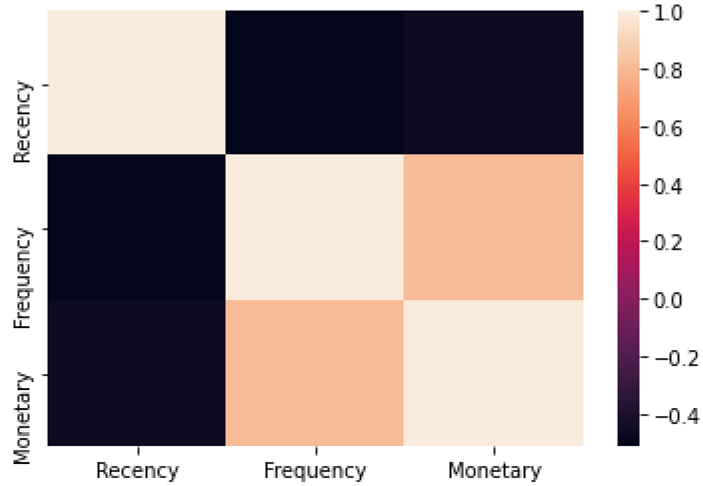


And a comparison between the data plotted in 2d (where the third axis is the color) before and after the log transformation:





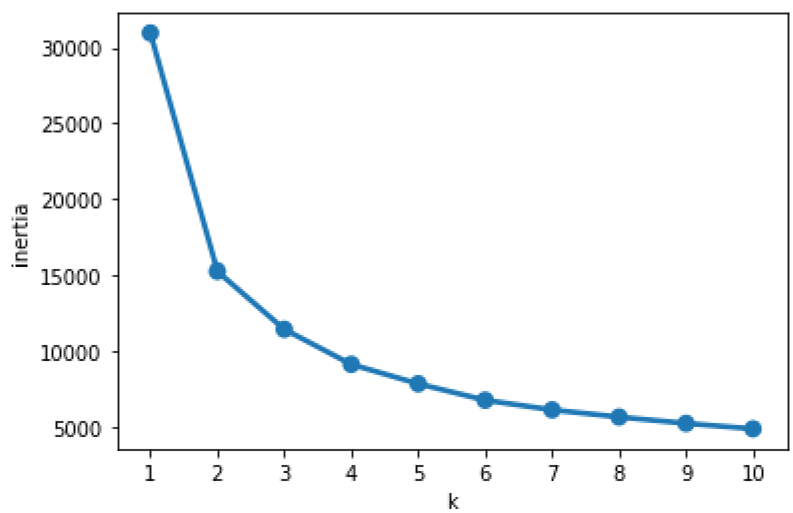
The latter plot is already suggesting that there should be roughly 4-5 clusters. We should check if there is any strong feature correlation:



Seems like Frequency and Monetary are correlated (which makes sense) but we should keep both since we understand the difference between them.

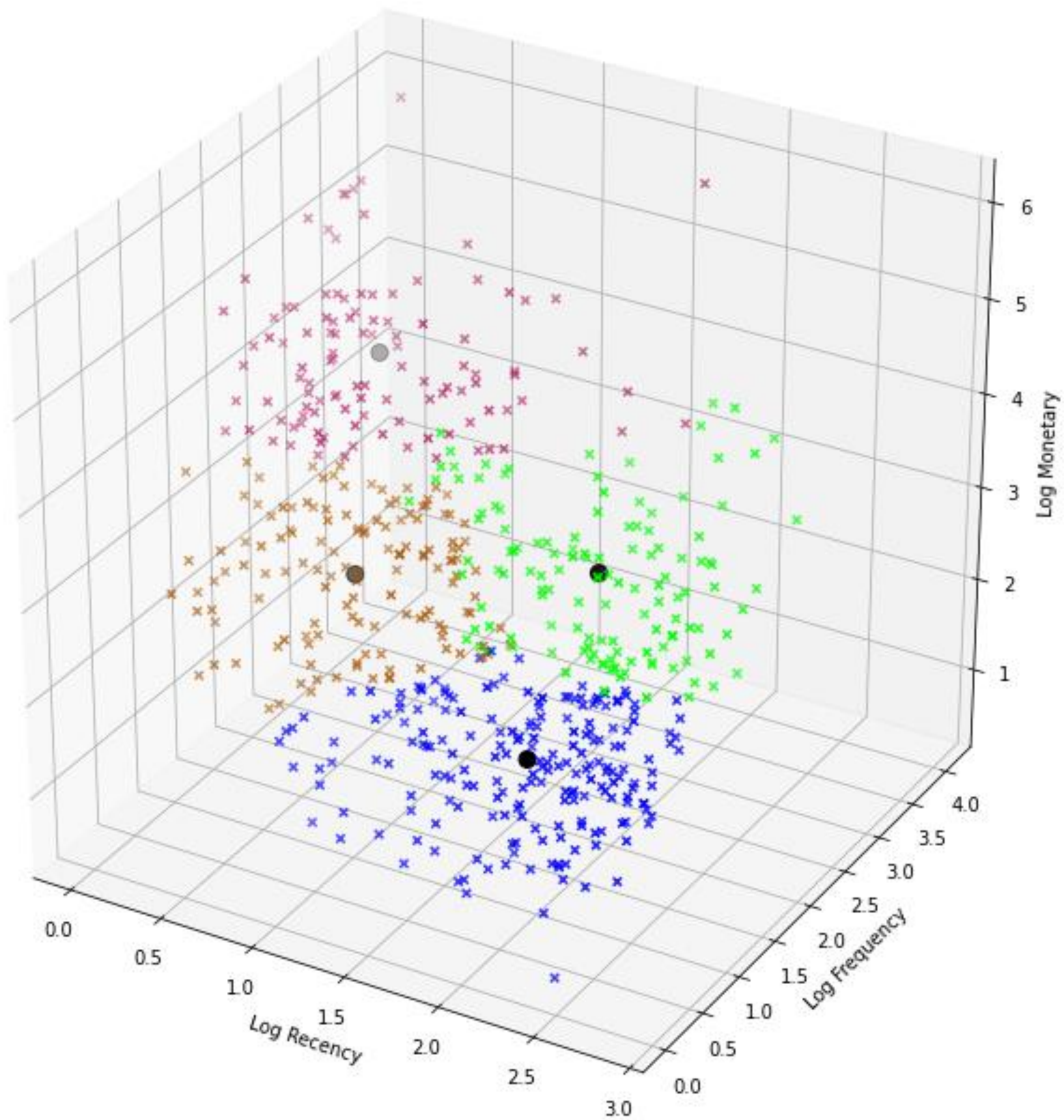
Modelling:

The following is a plot of the error of the k-means algorithm vs. k:



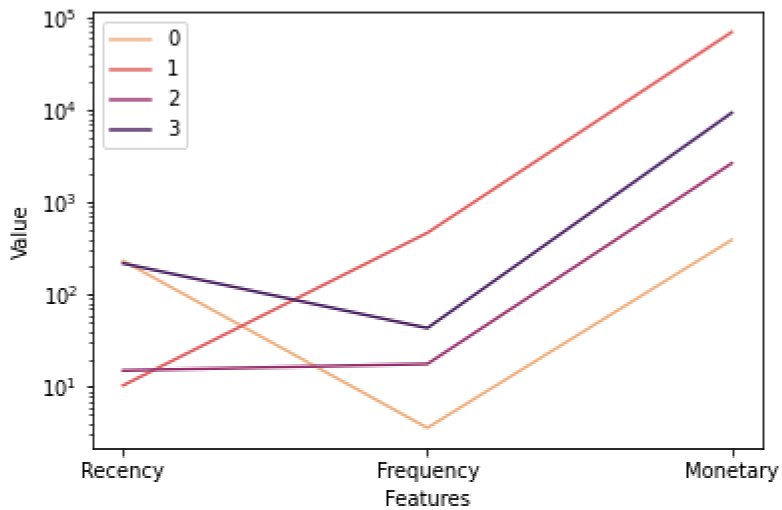
Using the elbow method we choose k=4 which confirms our intuition that came from plotting the data.

Here is a 3D plot of the clusters and their centers:



Results and Discussion:

	Recency	Frequency	Monetary	Cluster
Cluster				
0	232.848101	3.587223	392.237116	5056
1	10.348753	467.886881	69718.396979	2767
2	15.030303	17.700406	2651.202390	3201
3	218.172528	43.366396	9321.851178	3327



Based on the results we interpret the type of merchant of each cluster type:

Cluster	Merchant Type	RFM attributes
0	Churner merchant (lost)	Old last transaction, with low frequency and monetary.
1	Best merchant	Recent last transaction, with high frequency and monetary
2	New merchant	Recent last transaction, with low frequency and monetary.
3	At risk/churner merchant	Old last transaction, with moderate frequency and monetary.

- The scenario for merchant from cluster 0 might be something like: he tried the platform few times and never continued processing with it.
- For a merchant from cluster 3 we have a moderate frequency and monetary which means that he used the platform at some point in time. However, it might be that he churned, or it has been long since his last transaction due to long interpurchase time or other business reason.

<u>Merchant churn prediction</u>

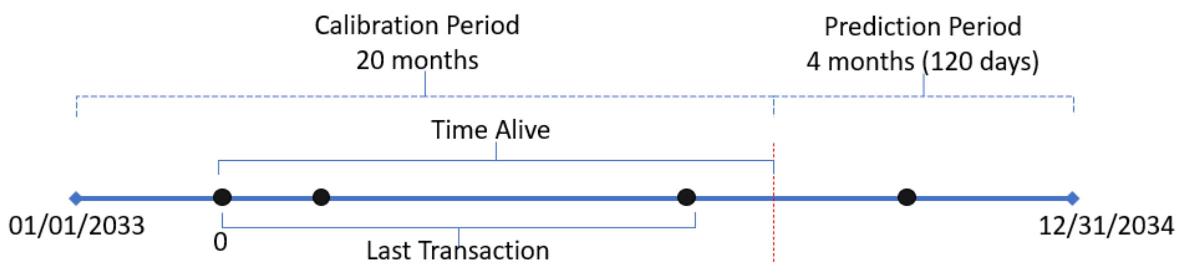
Method, Assumptions and Data Preprocessing:

A naïve way of predicting churners is to use the last discussion section and say that any merchant from cluster 0 has churned and any merchant that is in cluster 3 or moves to cluster 3 in the future will churn in the future but we want a prediction model that is much stronger than that (one limitation for example is that scores are not unique to an individual merchant but rather to the segment). The problem is that we do not have any labels for churner merchants, only transactional data so we need to come up with a definition of churners so that we could generate labels from the data we have so that we can train a supervised learning model.

The more common approach (see references) that we will use is to predict churn by training a **supervised algorithm** (e.g., random forest, logistic regression etc.). This approach provides greater granularity that the RFM method in that every merchant will have a predicted value unique to them based on their data. The target variable of course is binary (i.e., churned or still active) and the features are a snapshot of merchant attributes at some point in time. The main limitations of this approach are:

- Can potentially be more difficult to interpret by external stakeholders.
- Churn events are not modeled/studied over time (no censorship) – Survival Based models.

For model building purposes, we split the time window of the data into two consecutive subperiods called 'calibration' and 'prediction' respectively. **The duration of the calibration period is 20 months followed by the prediction period that is 4 months (120 days)**. This duration was chosen to capture activity/inactivity of customers with long interpurchase times (average time between transactions) and, at the same time, capture churn of those with short average interpurchase times as soon as possible.



The prediction period is set to be approximately equal to the average interpurchase time of a merchant in the 97.5% quantile. Meaning that 97.5% of the merchants have an average interpurchase time of less than 4 months (120 days). This approach enables us to avoid prediction periods being too long (and failing to detect churn long after it has happened), or too short (and misclassifying merchants with relatively longer interpurchase time as churners).

Definitions:

Churner (Merchant that churned) – coded as 1: A merchant that was 'active' during the calibration period (meaning it had at least one transaction during this period) that has no transactions during the prediction period.

Non-Churner – coded as 0: A merchant that was active during the calibration period that has at least one transaction during the prediction period.

- We do not include non-active calibration period merchants in our model which gives us 11,630 merchants out of the possible 14,351.

Features:

All features are a derived from a snapshot of the data at the end of the calibration period (we must only use historical data for future predictions!).

- Last Transaction (0<=Last Transaction<=Time Alive) – The number of days between the merchant's first transaction and the most recent merchant transaction date that is before the end of calibration period date.

- Time Alive – The number of days between the merchant's first transaction and the end of calibration period.
- Frequency - The total number of transactions before the end of calibration period.
- Monetary (in USD) – The total amount of dollars spent in all the merchant's transactions before the end of calibration period.

Note that the "Last Transaction" feature is very similar to "Recency". The difference is that it is measured from the merchant's first transaction and that it only refers to the transactions that are before the end of calibration period. The features were based on the RFM analysis we used above and we added the Time Alive feature since we want to try and teach the model how long we had observations for each merchant (the prediction should be different if the merchant only started using the platform and has few transaction versus a merchant that is alive for long but hasn't used the platform for a while).

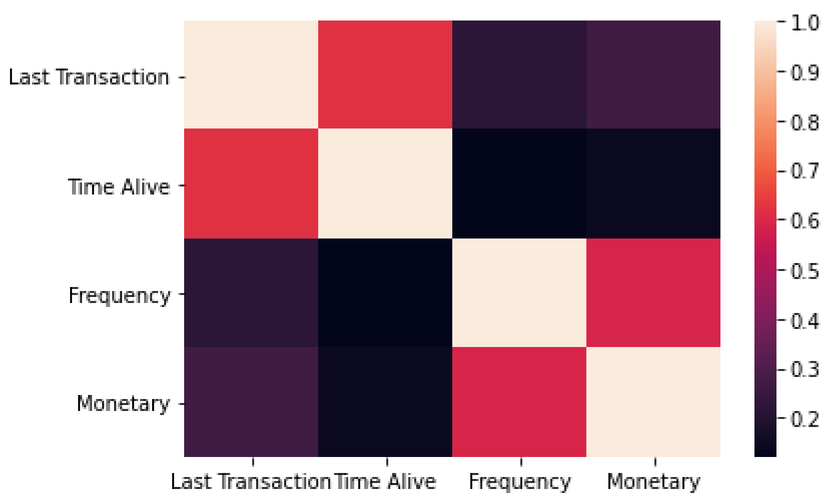
Labels (target variable): is being determined using the definitions above.

merchant	Last Transaction	Time Alive	Frequency	Monetary	Churn
0002b63b92	0	475	1	33.79	1
00057d4302	66	461	28	295.21	1
000bcff341	0	390	1	78.26	1
000ddbf0ca	0	458	1	102.99	1
000ed1585f	549	562	59	15754.72	0
...	...	...	...	...	...
ffd3e45675	23	607	5	726.26	1
ffe1f6b51a	260	456	53	2816.16	1
ffe26b900d	254	255	53	7164.12	0
ffec05edb9	20	221	3	159.34	1
fff1754102	382	391	41	4988.59	0

	Last Transaction	Time Alive	Frequency	Monetary	Churn
count	11630.000000	11630.000000	11630.000000	1.163000e+04	11630.000000
mean	165.279192	282.815219	89.671797	1.366230e+04	0.446776
std	169.446861	172.213088	458.658958	5.377734e+04	0.497180
min	0.000000	0.000000	1.000000	2.070000e+00	0.000000
25%	14.000000	133.000000	3.000000	3.392225e+02	0.000000
50%	105.000000	275.000000	10.000000	1.453295e+03	0.000000
75%	284.750000	428.000000	39.000000	7.376448e+03	1.000000
max	610.000000	610.000000	22350.000000	2.185458e+06	1.000000

- The percentage of labels labeled churner is 45% which means our model data set is balanced.
- The statistics of the RFM features of the dataset is similar to the original dataset even after removing the 2,721 users that were not active during the calibration period.

We should check if there is any strong feature correlation:



Seems like Frequency and Monetary are somewhat correlated as before (expected). Also, Last Transaction and Time Alive are somewhat correlated as expected.

We split the data into 80%-20% train, test sets randomly and standardize the data (mean=0 and standard deviation=1).

Modelling:

To this end we considered the following classifiers: Logistic regression, Kernel SVM, Random Forest and Gradient Boosting. Here are the results for their ROC AUC and accuracy:

	Algorithm	ROC AUC Mean	ROC AUC STD	Accuracy Mean	Accuracy STD
1	Kernel SVM	87.36	0.87	80.41	1.09
0	Logistic Regression	86.96	1.06	79.42	1.33
2	Random Forest	86.50	1.12	79.02	1.08
3	Gradient Boosting	86.39	1.02	78.60	1.20

Those results are close in performance, but we are mostly interested in minimizing false negatives as their cost is higher than the cost of false positives (since I believe that the platform would care more about a merchant becoming a churner that it missed than a merchant it contacted in fear it would churn that would stay active). So let us examine the results in terms of Precision, Recall and F1/2 score.

	Model	Accuracy	Precision	Recall	F1 Score	F2 Score
0	Logistic Regression	0.79	0.79	0.70	0.74	0.72
1	Kernel SVM	0.80	0.78	0.75	0.76	0.76
2	Random Forest	0.79	0.76	0.76	0.76	0.76
3	Gradient Boosting	0.78	0.79	0.68	0.73	0.70

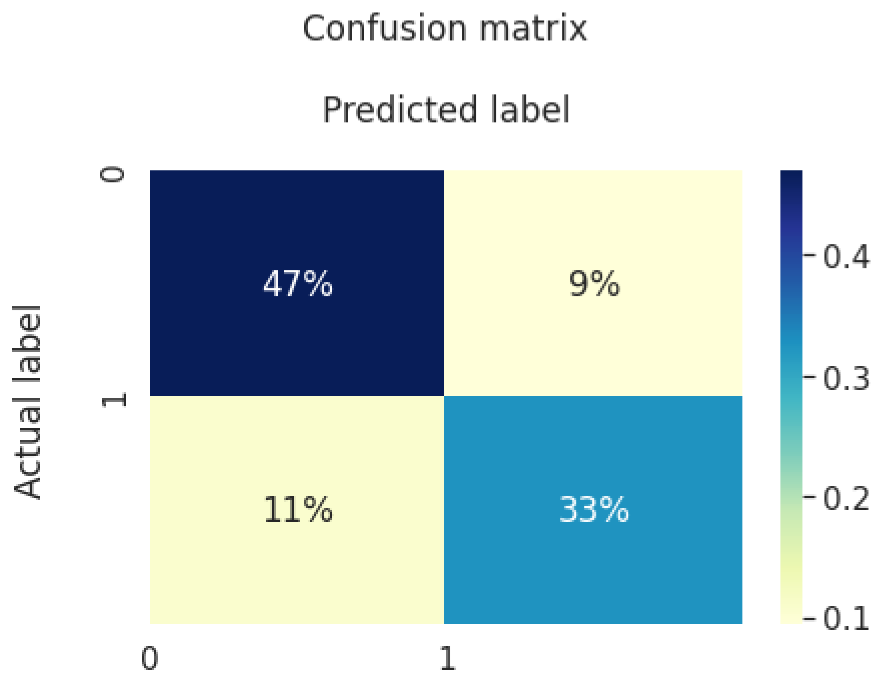
Based on this table and the latter one we chose Kernel SVM as our model.

Results and Discussion:

The 10-Fold cross validation of the classifier accuracy is:

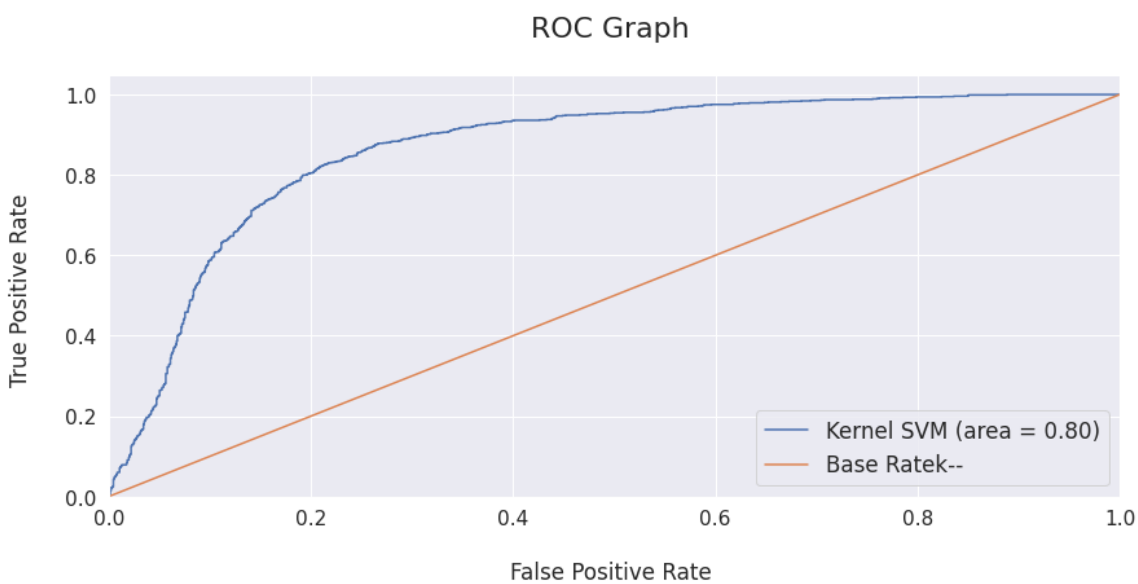
```
Kernel SVM Classifier Accuracy: 0.80 (+/- 0.02)
```

Confusion matrix:



Giving us a low false negative percentage of 11%.

And the ROC Curve:



Future Work:

- We can try and remove merchants from cluster 0 from model as they churned very fast and may not be indicative of an active merchant in the calibration period.
- We can try and tweak the calibration and prediction periods to get better classifiers.
- Try to fit a neural network classifier.
- Try to analyze the results with a business/marketing team that can have real world attributes input about merchants.
- Develop a survival-based model and compare to the supervised model.
- Develop an online learning model that learns after each transaction is logged.

References:

1. Managing B2B customer churn, retention and profitability, Ali Tamaddoni Jahromia,1, Stanislav Stakhovycha,2, Michael Ewingb,∗, Industrial Marketing Management, 2014.
2. Comparing Churn Prediction Technique sand Assessing Their Performance: A Contingent Perspective Ali Tamaddoni1, Stanislav Stakhovych2, and Michael Ewing, Journal of Service Research, 2016.