Elo Merchant Category Recommendation

Project proposal

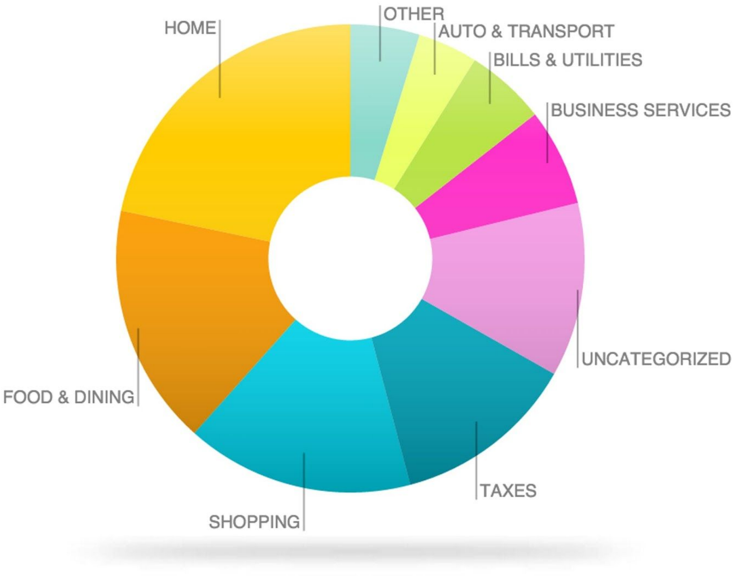
Can we predict customer’s loyalty score to help recommend merchant category to improve their shopping experience using data from their credit card transactions?

We rely heavily on reviews and feedbacks to learn customer’s satisfaction and that can easily predict their loyalty score. Credit card transactions are not emotional attached, hence predicting whether a customer will come back to the same merchants or the same category can be blindsided.

Elo has provided us with different files including historical transactions, new merchant transactions and merchants. However, they have normalized the purchase amounts.

Using feature_1, _2, and _3 to predict loyalty score. It verifies the importance of feature selections. What would be the best features to predict loyalty score would be the first question before looking at the loyalty score.

Let’s put the question this way: when looking at a customer credit card statement, what can we learn from their spending habit? One thing comes to mind: time stamp. Are they regular shopper or occasional shopper. Category: what kind of spenders they are.



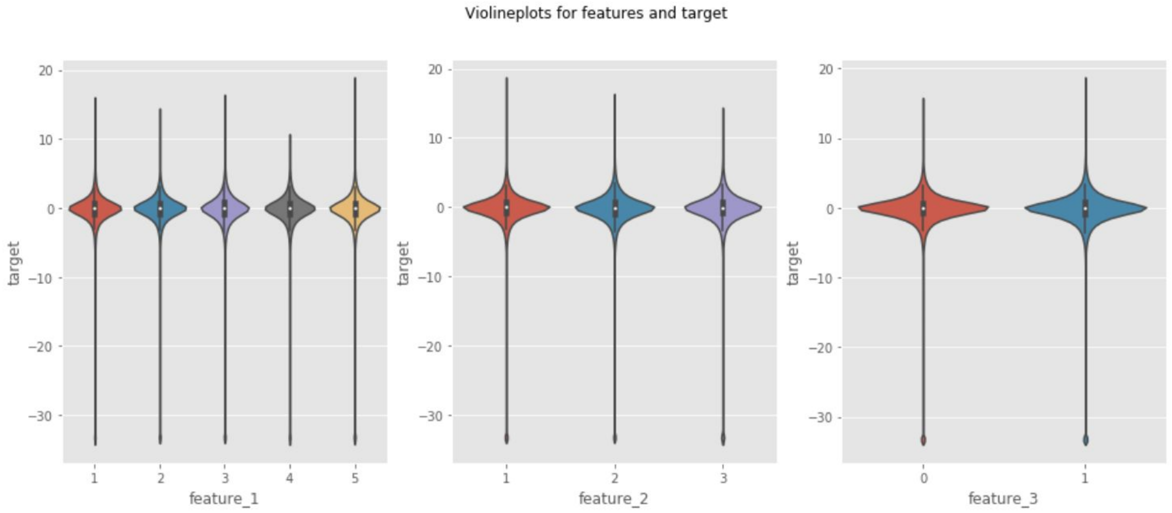
It is obvious what would be the most spending category, but if we know the most spending category, can we recommend the next category?

Exploratory Data Analysis Part 1: *descriptive statistics and visualization techniques*

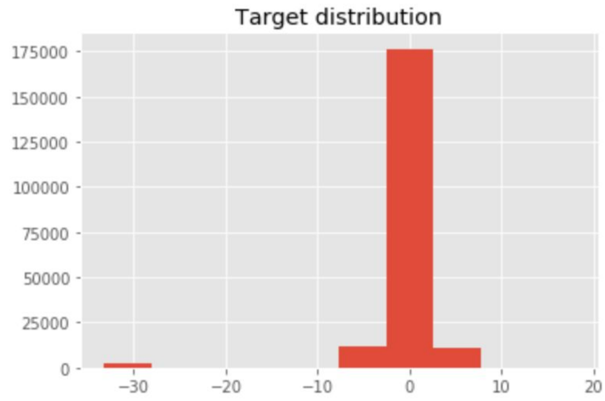
Dataset is obtained from Kaggle. This dataset is for Elo Merchant Category Recommendation Competition

There are 6 files in this dataset: train.csv, test.csv, sample_submission.csv, historical_transactions.csv, merchants.csv, new_merchant_transactions.csv.

train.csv - the training set which we will use to train our model. It has 1 date column, 3 feature columns with categorical values, and 1 target column. The violine_plots show normal distributions, we can apply this to our inferential statistical tests in later parts. There are 201917 rows in the training data set.

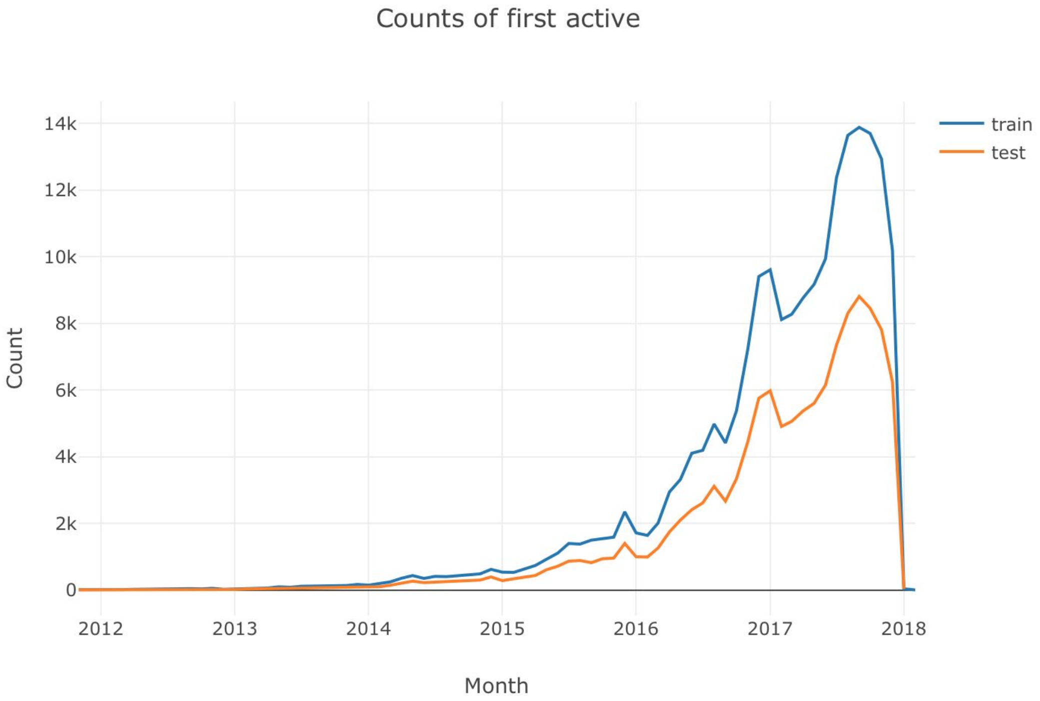


Checking with the target distribution, it is also normally distributed, however, there seems to be some outliers at -30: 2207 samples, which will need further analyzing.



test.csv - the test set which we will use to test our model. This has a similar format with train.csv, except it does not have a target column, which we need to predict. There are 123623 rows in the testing data set.

There is a similar trend in counts of first active month. But there is a steep decline at the end. It might be a change of cards, an update in the new system, or something.



sample_submission.csv - a sample submission file in the correct format - contains all card ids we are expected to predict for. This is because we need a standardized form to rate/score the submissions.

historical_transactions.csv - up to 3 months' worth of historical transactions for each card_id. Using .head() we are able to get the first 5 lines and all the columns of the dataframe. One thing we do not expect were that 3 features in the training and testing datasets are not in this file. Unlike the first two datasets, this one needs a lot of wrangling due to its many more features are object types.

merchants.csv - additional information about all merchants/ merchant_ids in the dataset. There are some missing rows in this data set.

new_merchant_transactions.csv - two months' worth of data for each card_id containing ALL purchases that card_id made at. This has the same format with historical_transactions.csv.

There is a dictionary excel file giving definitions of the data fields.

Data Wrangling

Training dataset and sample submission: No further wrangling

Testing dataset:

There is one line of missing data in the test dataset. We fill it in with the first data, having the same values for features.

Historical_transactions:

We convert authorized_flag into a binary value: 1 and 0 instead of N and Y respectively.

Installments are then converted into categorical. Purchase date changes its format to datetime.

For category_1, we convert Y to 0 and N to 1, for category_3, we convert to A to 0, B to 1, C to 2, and nan to 3 by mapping them out. We do not convert category_2 because it is in 1-5 already.

New_merchants_transactions: similar methods with historical_transactions:

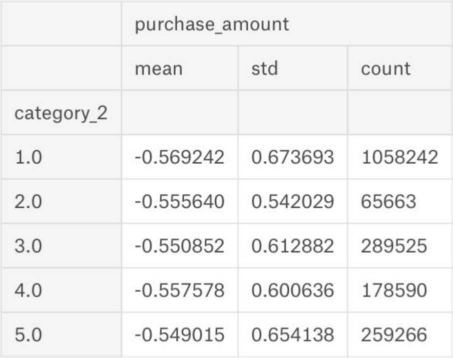
Merchants: There are missing values in merchants.csv

For category_1 and category_4, we convert into binary values, Y as 0, and N as 1 , for category_2, we fill nan with 0.

Aggregate transactions: to further understand the data

We separate the transaction date into year, month, days of week, weekend, hours. For each columns we then have aggregations of mean, sum, average, max, min, standard deviation respectively.

For example: Here we look into purchase amount of category 2 of new_merchants_transactions. By separating the dataset into different categories using groupby, we can take a closer look into the data.



	purchase_amount		
	mean	std	count
category_2			
1.0	-0.569242	0.673693	1058242
2.0	-0.555640	0.542029	65663
3.0	-0.550852	0.612882	289525
4.0	-0.557578	0.600636	178590
5.0	-0.549015	0.654138	259266

Exploratory Data Analysis Part 2: *inferential statistics techniques*

Before getting to hypothesis, we will take a look into statistics: mean, min, max etc.

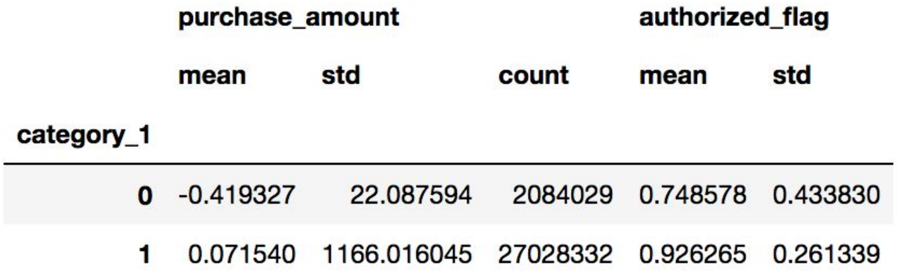
Historical_transactions:

91.3545% of transactions are authorized. Since the flagged rate is pretty low here. We can ignore this since analysis is about loyalty score not fraud detection.

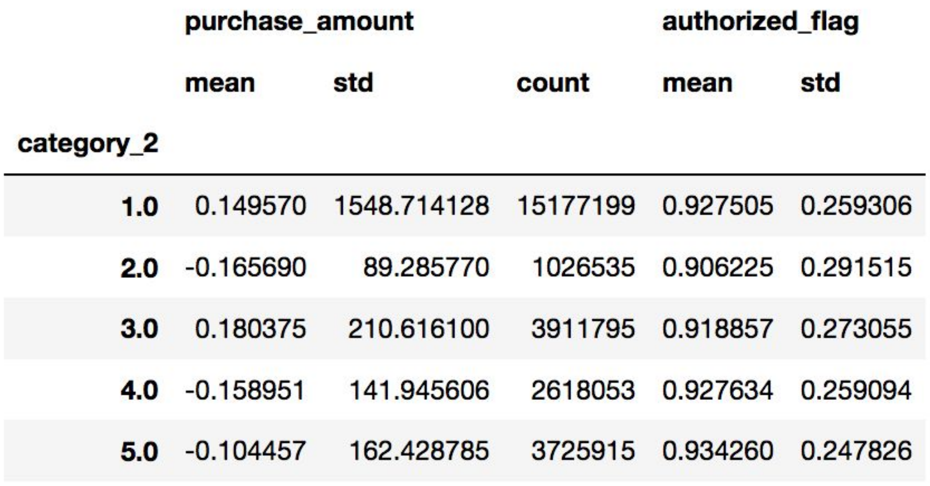
One current trend with financial institutes in America is installment shopping where customers can pay in desired installments for a fixed fee instead of the traditional credit card fee that tends to accumulate interests. However, Elo is in Brazil. With the most common installment as 0, Brazilians tends to pay at once. Is this because customers do not need installments, or that installments are not provided by merchants? There are 2 outliers of -1 and 999 installments, which could have been used to fill missing values.

Purchased amounts are normalized. One would have think that purchased amount could significantly helps predict the loyalty score. By normalizing them, the prediction task is now more difficult.

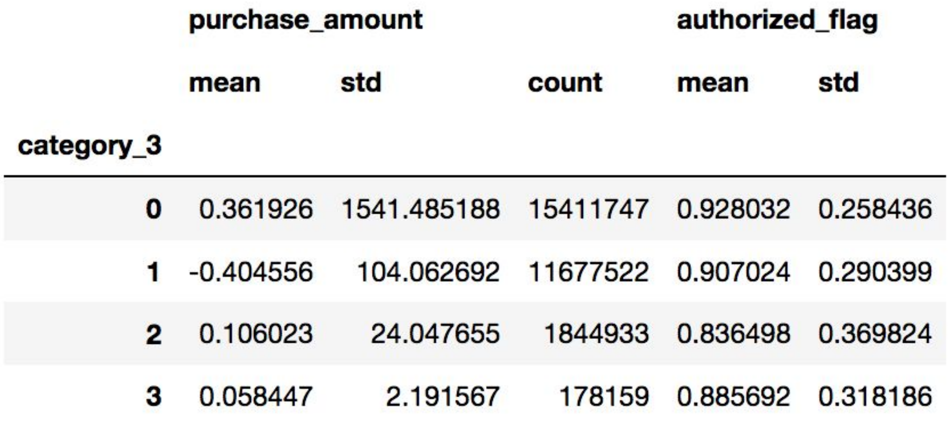
We now group them historical transactions by categories:



	purchase_amount			authorized_flag	
	mean	std	count	mean	std
category_1					
0	-0.419327	22.087594	2084029	0.748578	0.433830
1	0.071540	1166.016045	27028332	0.926265	0.261339



	purchase_amount			authorized_flag	
	mean	std	count	mean	std
category_2					
1.0	0.149570	1548.714128	15177199	0.927505	0.259306
2.0	-0.165690	89.285770	1026535	0.906225	0.291515
3.0	0.180375	210.616100	3911795	0.918857	0.273055
4.0	-0.158951	141.945606	2618053	0.927634	0.259094
5.0	-0.104457	162.428785	3725915	0.934260	0.247826



	purchase_amount			authorized_flag	
	mean	std	count	mean	std
category_3					
0	0.361926	1541.485188	15411747	0.928032	0.258436
1	-0.404556	104.062692	11677522	0.907024	0.290399
2	0.106023	24.047655	1844933	0.836498	0.369824
3	0.058447	2.191567	178159	0.885692	0.318186

We could not observer any patterns in these categories. But perhaps clustering will show?

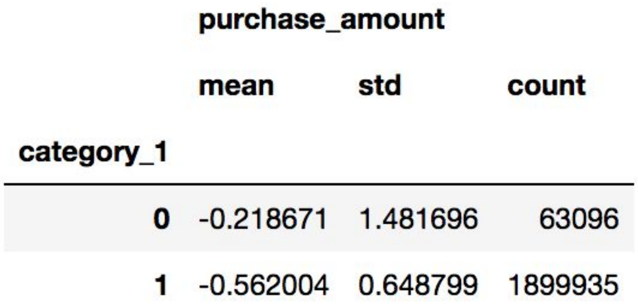
The transactions were recorded in 308 cities in 25 states with 327 different merchant category by 326311 merchants in 41 different subsectors. The data is broad and diverse.

new_merchant_transactions: using the same step as above with historical_transactions

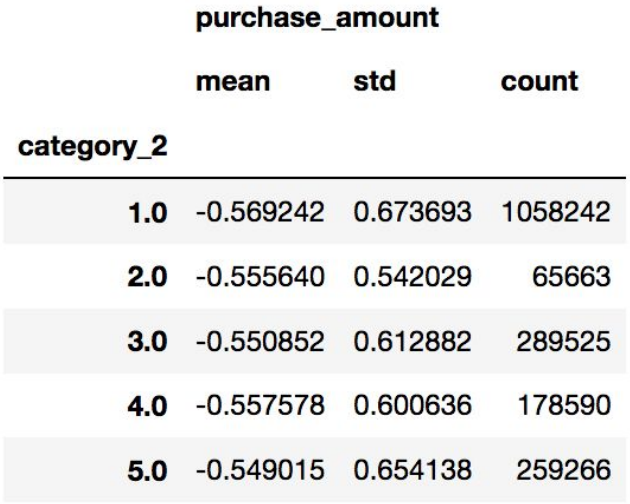
With an impressive 100% transactions are authorized. Is there a significantly improved electronic payment system?

There is an observed trend here: 0 installment : 922244 and 1 installment : 836178 more transactions now have 1 installment comparing to more transactions paying up front in historical transactions(0 installment: 15411747; 1 installment:11677522) The US installments trend are caught up in Brazil.

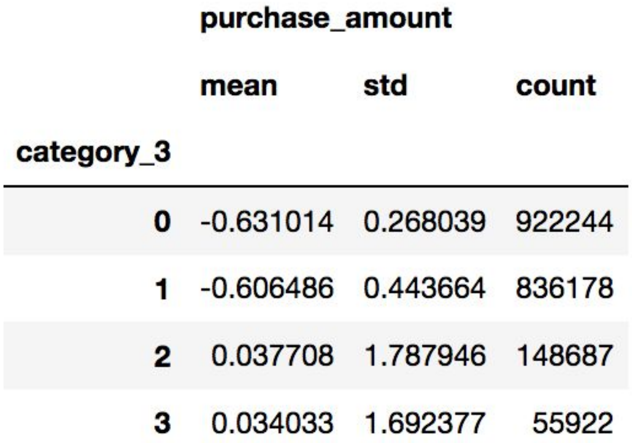
Since 100% transactions here are authorized, our categories tables now won't have the flagged section:



	purchase_amount		
	mean	std	count
category_1			
0	-0.218671	1.481696	63096
1	-0.562004	0.648799	1899935



	purchase_amount		
	mean	std	count
category_2			
1.0	-0.569242	0.673693	1058242
2.0	-0.555640	0.542029	65663
3.0	-0.550852	0.612882	289525
4.0	-0.557578	0.600636	178590
5.0	-0.549015	0.654138	259266



	purchase_amount		
	mean	std	count
category_3			
0	-0.631014	0.268039	922244
1	-0.606486	0.443664	836178
2	0.037708	1.787946	148687
3	0.034033	1.692377	55922

No pattern in 3 categories, and no comparable pattern between historical and new transactions.

Transactions are recorded in the same cities, states, and have the same sub sectors when there are only 314 merchant categories and 226129 merchants.

Merchants:

There are 334696 merchants, 324 merchant categories, and 109391 merchant groups.

Besides categorical features like transactions, merchants data have numerical features that we need to look at one by one.

Recall the features in the training set were normally distributed, the numerical_1 is skewered with 68% to the left, and the numerical_2 is skewered with 74% to the left. In fact these two are very similar in distributions. Only about 10% of the merchants have different numerical_1 and numerical_2.

Similarities can be observed in recent sales range and recent purchase range as well.

Since the features in merchants are skewered, we decide to not take into account of the merchants file in modeling.

Next part will be preprocessing for modeling.

Link to EDA jupiter notebook on github: