Predicting Loyalty Scores Based on Historical Credit Card Transactions

Caitlin Hung, Michael Liu, Soheil Karima

***Abstract*—This paper is a brief report about our exploratory search on the relationship between credit card past purchasing behavior and its associated customer loyal scores. By applying various machine learning models trained with the ELO customer loyalty dataset, we identify key features on predicting customer loyalty scores and provide comparisons on performance among distinct machine learning techniques.**

I. Introduction and Relevant Literature

BUSINESSES have historically tried to collect and analyze customer information to manage their relationship with customers. Understanding the relationship between customer behavior and customer retention is critical for engaging with customers, co- creating value, and strengthening customer loyalty, which is essential for business growth. Thus, many businesses have also explored the use of machine learning techniques in improving this understanding [1]. A familiar approach to the problem is the use of collaborative filtering. Much like building a recommender system for movies, music, or likely items to be purchased, the idea behind predicting customer loyalty with collaborative filtering is to find known customers (or card IDs) with similar behavior and predict like-values for the unknown. Common techniques for measuring similarity include cosine, Pearson, and Euclidean measures between extracted features. Different clustering algorithms such as k-nearest neighbor or k-means then group together similar customers and predict values based on the result. Notable companies such as Netflix and Amazon use collaborative filtering to personalize their users experience [2]. A less familiar approach is the use of recurrent neural networks (RNN). A research paper from KTH Royal Institute of Technology explored using RNNs capacity for memory to make future CLV predictions from a time series of current Customer Lifetime Values (CLV) data. CLV is closely related to customer loyalty, as it assigns a monetary value to a customer relationship as a measure of churn, which is used to identify customers that become less loyal. The experiment concluded that RNN is a promising approach, as it was capable of identifying trends among members, although further exploration is necessary to produce more useful, accurate results [3].

II. Dataset and Exploratory Analysis

We chose a synthetic data set provided by Elo, one of Brazil's largest payment brands. The data simulates credit card transaction history and merchant information. It contains approximately 30 million transactions between a card and merchant ID, along with associated information like date, city, and normalized amount of purchase for that transaction. Additionally, it includes over 300,000 merchant details, like merchant category groups, the monthly average of transactions for different periods of time, and ranges for revenue and quantity of transaction in the last active month. Finally, we are given 325,540 unique card IDs, about two-thirds (201,917) of which have loyalty scores provided. One noted challenge of this dataset was the presence of anonymized categories and measures, so in determining feature importance, it was critical that we visualize the distribution of those values to better understand how to incorporate them into our understanding of the dataset. Following, we provide greater detail about the data associated with cards, merchants, and transactions as we describe our exploratory analysis of the dataset. Each card ID is linked to anonymous features 1, 2, and 3. The features take a single value between 1 and 5; 1 and 3; and 0 and 1, respectively. On exploration, we observed that all three features have similar loyalty score/target distributions regardless of the value they take. Figure one contains box plot visualizations of the loyalty score distribution for each of the three features. On calculating statistics for the matching features of the test set we determined that test and training set distributions are very similar, with mean, median, and standard deviations within two significant figures of each other. Figure two illustrates the distribution of counts for the first active month of card IDs in the train and test sets. Note that this is not a stacked histogram, but that there are about twice and many cards in the training set as the test set. Finally, we performed rough visualization of linked card and transaction data. Figure three shows associated category distribution and normalized purchase amounts over time for a randomly selected card.

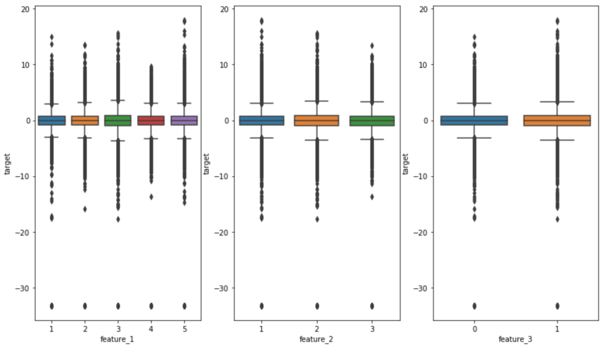


Fig. 1. Visualization of anonymous features to target loyalty score in training and validation set.

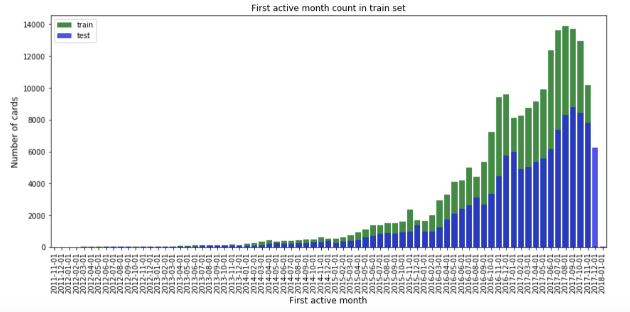


Fig. 2. Visualization of first active month counts in train and test sets.

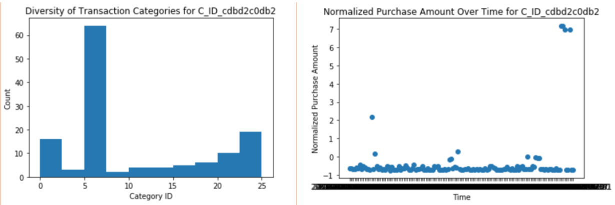


Fig. 3. Rough visualization of transaction data for a randomly selected card ID.

III. Predictive Task, Features, and Evaluation

The predictive task that we focus on is determining the loyalty scores for card IDs. As defined by Elo, loyalty score is a ”business metric that considers both future spending and retention as main components”. Card loyalty (and by extension, customer loyalty) is important for any business as new customers tend to be more costly to acquire. As a result, customer loyalty is crucial for business growth. Improving our understanding of the relationship between customer behavior and loyalty score will allow businesses to determine future customer behavior and retention rate, which are used as the main metrics for creating the right experience for its users. We trained our model on approximately 100,000 provided card IDs and loyalty scores, tuned it on another 100,000, and tested it on an additional 100,000. Tuning and evaluation was done using RMSE, which is an appropriate measure given that predicting values that are very different from actual loyalty score could be costly in a business model, so we want to penalize large error.

$$\text{RMSE} = \sqrt{\frac{(X_{\text{pred}} - X_{\text{true}})^2}{N}}$$

Since the provided loyalty scores displayed a Gaussian distribution in our data visualization, we used predicting scores of 0 as our baseline. While all zeroes is a trivial predictor, it minimizes large error with normalized scores.

A. Feature Extraction with Data Aggregation

Our primary data exploratory goal is to extract important features from millions of historical transactions of individual credit cards. Because from the transaction table each credit card is associated with multiple transaction records and each entry in our train data is associated with one unique credit card id, our features are generated through aggregating information related to each unique credit card id. During the feature generation, we perform one hot encoding for merchant category id and subsector id.

Under the assumption we believe that customer loyalty is primarily determined by the period of active usage, the volume of purchases and the diversity of card usages, we are able to aggregate useful features (shown in Table.I) from the historical transaction dataset.

TABLE I
Extracted Features From Historical Transactions

Name	Description
purchase amount std/mean	two statistical summaries of the historical purchase amounts
purchase amount min/max	maximum and minimum of the historical purchase amounts
purchase date std/mean	two statistical summaries of the historical purchase dates
purchase date min/max	the oldest and the most recent historical purchase dates
merchant category ids sum	the proportional sum of merchant category one hot encoding
subsector ids sum	the proportional sum of subsector one hot encoding
frequency	the total number of historical purchases
month lag mean/std	statistical summaries of the month lag feature
month lag max/min	maximum and minimum of the month lag feature
card activation period	the period of card in active usages
card first purchase	the period between now and the first transaction occurred
feature 1	anonymized card feature for each card id
feature 2	anonymized card feature for each card id
feature 3	anonymized card feature for each card id

B. Feature Selection with Principal Component Analysis

Principal Component Analysis is a dimensional reduction technique that preserves the reconstruction errors through selecting the dominant feature vectors as principal components of the original data set. By inspecting these principal components through the technique, we are able to understand what are the major features that represent our train data set, and how they interact with the target customer loyalty scores.

$$x \approx \sum_{j=1}^{K} \phi_j y_j + \sum_{j=K+1}^{M} \phi_j y_j$$

By utilizing this dimensional reduction technique (PCA, Principal Component Analysis) on the joined feature set, we are able to have the crucial insight on the quality of our extracted features and how they interact with each other to contribute to the final customer loyalty scores. As the singular values of the first two components have suggested (shown in Fig.4), the dispersion of the train data points are largely demonstrated by the first two principal components. Because each principal component is a vector of a total combination of coefficients for all features, we are only displaying in our table the top 8 feature parameters with the largest absolute coefficient values from the first two principal components.

In Table.II at the month lag sum, which implicates the variance of purchase dates in referencing with Feb 2018, and

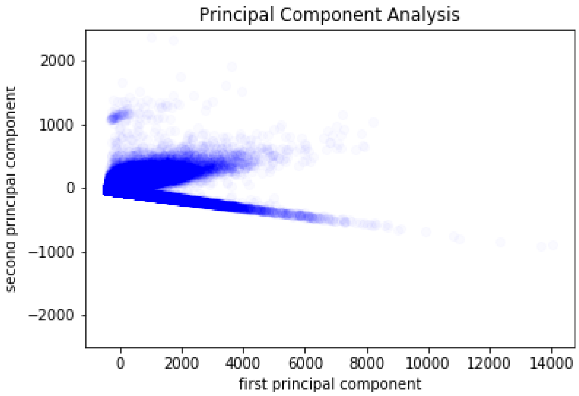


Fig. 4. Visualization of the train data points projected on its first and second principal components.

TABLE II
PRINCIPAL COMPONENTS OF FEATURE SET

p_1	v_1	p_2	v_2
month lag sum	0.98542	installments sum	0.98629
number of purchases	0.15732	number of purchases	0.10455
installments sum	0.06461	installments max	0.09660
month lag min	-0.00302	purchase amount max	0.01011
purchase amount max	0.00125	installments mean	0.00638
month lag std	0.0.00083	purchase amount mean	-0.0048
installments max	0.00058	month lag min	0.00269
card activation duration	0.00047	feature 2	-0.0020
card first purchase	0.000162	feature 1	-0.00137

number of past purchases during the past customer credit card behavior have crucial influence on the customer loyalty scores in addition to card usage period and statistics related to purchase amount and purchase frequency. Individual merchant category statistics and statistics related to various sub-sectors have less influence in the target values. Overall, by looking at all the top 20 principal components, the dimension reduction analysis reaffirms our assumption that customer loyalty score is largely influenced by the three attributes (from most important to least):

1) the diversity of customer's usages and longevity of the usages (purchase date)
2) the frequency of the card usages
3) the volume of the customer purchases (purchase amount)

IV. PREDICTIVE MODEL

A. Baseline Model

According to ELO's description of the customer loyalty train data set, the target values are given based on a normalized distribution. However the baseline predictor according to ELO is a predictor that predicts all zeros for all the data points on the test set. So we submitted the recommended baseline where we predicted only zero for every card ID in the test set and recorded the RMSE result as our initial baseline (i baseline). Then since the targets are normalized and the measurement is the RMSE value, therefore, it is reasonable for us to use a model which predict the global average customer loyalty score for all entries as our baseline model. So we submitted the global average for every test entry and that gave us a better RMSE result than just predicting zeros and we called this the advance baseline (a baseline). Throughout all of our efforts, we compared the result of our models on the validation set with the result achieved by initial baseline and advanced baseline models. The RMSE results of our final model and the two baseline models are demonstrated in III

TABLE III
TEST RMSE RESULTS

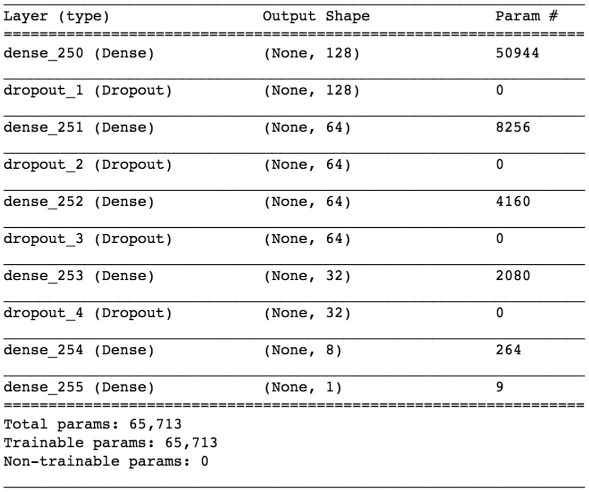
model	i baseline	a baseline	Deep Neural Net
RMSE On Test	3.952	3.930	3.888

B. Deep Neural Network

We decided to design a deep neural network which would help us extract the features from the given data-set (ELO). The given data-set contains many anonymous categories that we could not simply extract features from by simply graphing the data. So using a deep neural network which would extract features from the given input data is a great help to us to extract right features from the anonymous categories. Another reason for selecting a neural network model was the fact that we could model a non-linear predictor which we thought would fit the data better than a simple linear model. We tried multiple different structures of the neural network with a different number of hidden layers, number of cells at each layer and finally different activation functions such as (ReLU, LeakyReLU, and etc). Our inputs are 1 dimensional vectors of size 397 as described in the Table.I.

The layout of the model is presented in Fig.5. Since we are dealing with large input vectors and largely hidden layers we added dropout layers to reduce over-fitting in a neural network by preventing complex co-adaptations on training data. Even though with larger layers and more number of layers we could have achieved a better accuracy, we did not increase the depth of the network since that would have slowed down the process and we did not have the required computation power (GPU) to teach such model in a reasonable time frame. Another method we used to prevent our model from over-fitting is our performance comparison on the train and validation set. The strength of using a model like Deep Neural Network was that we did not need to worry about the complexity of the problem and we could rely on the model to figure out the complexity of the features, however this also made it harder for us to analyze and understand the reason behind the predictions made by the model.

We shuffled the provided data and split it in half for where we used the first half as the training set, and the second half as the validation. We used the validation set primarily for early stopping to prevent over-fitting and also to determine which model structure works the best. After picking the best structure based on the validation loss We let the model run for 200 epochs with batch sizes of 256. The result of the training and validation on the final model through the 200 epochs is illustrated in Fig.6.



```
Layer (type)                 Output Shape              Param #
=================================================================
dense_250 (Dense)            (None, 128)               50944
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_251 (Dense)            (None, 64)                8256
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_252 (Dense)            (None, 64)                4160
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_253 (Dense)            (None, 32)                2080
_________________________________________________________________
dropout_4 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_254 (Dense)            (None, 8)                 264
_________________________________________________________________
dense_255 (Dense)            (None, 1)                 9
=================================================================
Total params: 65,713
Trainable params: 65,713
Non-trainable params: 0
_________________________________________________________________
```

Fig. 5. Summary of the Deep Neural Net model structure

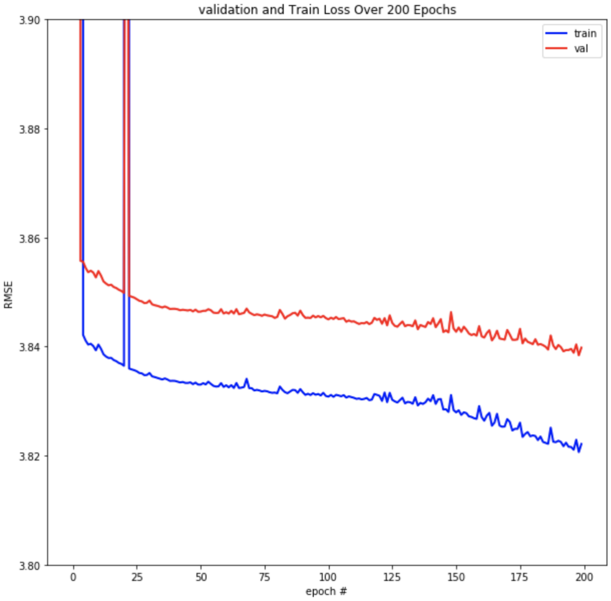


Fig. 6. Train and Validation loss converging through 200 epochs of training

C. Other Methods

1) **XGBoost**
eXtreme Gradient Boosting is considered one of the most popular technique for regression task since the algorithm is compatible with various type of feature data. The algorithm is known for dealing with structured financial data because of its utilization of tree regressors which is different from the nonlinear regression model we have implemented with neural network, and since our feature vectors have demonstrated significant relationship among themselves, it is reasonable for such method to successfully accomplish the regression task. However, the downside of such algorithm is its demand for the right hyper-parameters. In our experiment, tuning the various hyper-parameters: primarily the number of estimators and max depth per regression tree are quite difficult since each iteration is computationally expensive and time-consuming.

TABLE IV
XGBoost RMSE

Number of Estimators	100	120
Max tree depth	5	8
Validation RMSE	3.84273	3.83548

2) **K Nearest Neighbor** Considering that the data points are spread all over the feature space, we decided it might be a good idea to try K Nearest Neighbor. Using K Nearest Neighbor algorithm for each data point we would select the target of all K Nearest Neighbors and take the average of the targets and we assigned the average to be the target of the data point. We ran the algorithm with different values for K and selected the one with the least RMSE error. The results are illustrated in Table.V. The advantage using K Nearest Neighbor model is that the algorithm is able to identify different data clusters (in this case, the representative groups of credit customers). However, one of the major downside for implementing K-Nearest Neighbor is that it is not a learning algorithm such that it doesn't reduce reconstruction error during training phase.

TABLE V
K Nearest Neighbor Regression RMSE

$K =$	1	2	3	4	5	6	7	8	9
RMSE	5.40	4.72	4.46	4.31	4.24	4.18	4.13	4.10	4.08

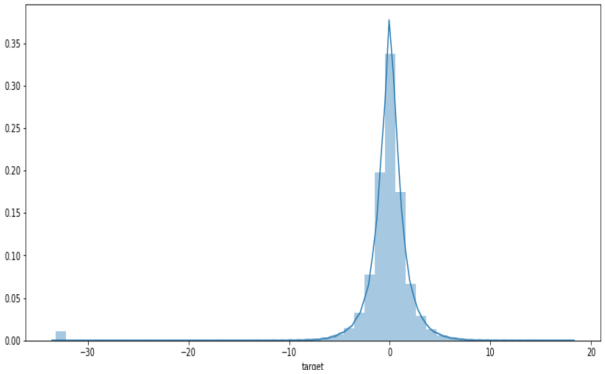


Fig. 7. Loyalty score distribution of our training and validation set.

V. Conclusion

To conclude, we are able to discover key features that significantly influence the relationship between past credit card purchasing behavior and customer loyalty scores by using feature aggregation techniques and dimension reduction techniques. With these significant financial indicators, we follow model selection principles to experience with various machine learning algorithms and inspect each of the performances. Comparing the experimented models, deep neural network occurs to us as the more appealing algorithm since it is both computationally effective and particularly suitable for regression related tasks. Although we have also experienced with XGBoosting techniques and KNN algorithms, in the future, our research can potentially be further using the state-of-the-art technique in financial modeling such as Random Forest and automatic relevance determination neural networks etc.

References

[1] Aluri, Aj Price, Bradley Mcintyre, Nancy. Using Machine Learning to Cocreate Value through Dynamic Customer Engagement in a Brand Loyalty Program. *Journal of Hospitality Tourism Research.109634801775352. 10.1177/1096348017753521.*, January 2018.

[2] Chhavi Saluja. Collaborative Filtering based Recommendation Systems exemplified. https://towardsdatascience.com/collaborative-filtering-based-recommendation-systems-exemplified-ecbffe1c20b1.

[3] Jesper Ljungehed. Predicting Customer Churn Using Recurrent Neural Networks. *KTH Royal Institute of Technology*, June 2017.