ABSTRACT

The estimation of Customer Lifetime Value (CLV) is one of the core pillars in strategy development and marketing. CLV is a measurement in dollars associated with the long term relationship between a customer and a company, revealing how much that customer is worth over a period of time. CLV is rather predictively powerful when considering customer acquisition processes, as well as for selecting optimal service levels to provide different customer groups.

Current methods for estimating CLV involve building a single model using the entire population of customers as input. This not only loses the granularity in the data, but also gives rise to poor targeting and strategic advertising for consumers. This paper seeks to show the advantages of combining smaller, targeted models intelligently in order to build separate models for customers buying different product categories. This enables an effective use of the data that firms have about customers to make intelligent strategic decisions.

Using a sample dataset, the proposed implementation of Multitask learning ([2]), in which knowledge learned is shared between related categories, yields a strong improvement in CLV forecast accuracy as compared to using a single, large model on product categories with lower numbers of transactions (less than 150). Additionally, this same method reduces the standard deviation of error when compared to the single large model. Most significantly, the Multitask learning models tend to perform better than single models when categories have sparse data to train on, traditionally considered a harder task. These results indicate that Multitask learning techniques can lead to a better outcome than current industry standards, and perhaps is a better alternative to the existing methodology.

1. INTRODUCTION

In marketing, Customer Lifetime Value (CLV) is a prediction of the net profit attributed to the entire future relationship a company has with a customer. Accurate and timely calculation of CLV is important because knowing which customers are likely to be very valuable to any given company, as well as those which will cause the firm to lose money, can help greatly with tailoring the marketing, product offerings, and the purchasing experience to specific customer segments in order to maximize revenues and profits [15]. CLV calculations allow firms to build a picture of which type of customers are valuable or under which circumstances they become valuable. This can drastically improve the value of money spent on customer acquisition and marketing. This paper provides a method for using CLV to predict which product categories are valuable, which is a particularly difficult estimation in new categories with sparse data.

This paper proposes a model for calculating CLV without losing the granularity in the data, unlike other models. Whereas many methods used today take all of any given customer’s transactions into account when calculating their CLV, the proposed model will subdivide each customer’s transaction by product category, followed by calculating each customer’s CLV specific to any given product category. For example, rather than calculating the CLV of a customer who has bought both televisions and cameras, the proposed model would calculate a separate CLV for that customer for each product category (the total CLV would then be the sum of those two CLVs). This, however, introduces a problem: since each category will have less data to build a model upon than the original model, some category’s models will be weak and largely overfitted if their data is used in isolation.

To solve this problem without compromising the advantages of tuning models to specific categories, the model will incorporate Multitask learning. Multitask learning is a branch of machine learning consisting of several algorithms for sharing knowledge between different tasks. Product categories which have fewer transactions (and therefore inherently weaker models) will be bolstered by information gained from more popular and related product categories. The proposed model would calculate a separate CLV for that customer for each product category (the total CLV would then be the sum of those two CLVs). This, however, introduces a problem: since each category will have less data to build a model upon than the original model, some category’s models will be weak and largely overfitted if their data is used in isolation.

To solve this problem without compromising the advantages of tuning models to specific categories, the model will incorporate Multitask learning. Multitask learning is a branch of machine learning consisting of several algorithms for sharing knowledge between different tasks. Product categories which have fewer transactions (and therefore inherently weaker models) will be bolstered by information gained from more popular and related product categories. This method leaves popular categories’ models relatively unchanged, while greatly strengthening less popular categories’ models. In short, the proposed model is designed to address the following problem statement:

Given sample transactions, predict Customer Lifetime Value at the product category level, even if data per category is sparse.

The model will take as input a predefined transaction log,
and it will output a CLV for each customer in each product category they have made purchases in. Additionally, useful graphics and other visualizations are produced based on the output to make the data more accessible and useful in deriving key insights.

This document examines related efforts at solving this problem, then outlines the schema and software implementation of the proposed system, and finally discusses the results of the system on sample data and looks towards potential improvements. First, it reviews the current research in calculating CLV so as to show how this approach is novel and useful. Second, it describes the proposed model and its corresponding step-by-step implementation of an algorithm to predict CLV using Multitask learning. Then, it expounds on the evaluation criteria to be used to determine the success of this project and how well the model compares to these criteria. Finally, suggestions for future improvements or expansions are put forward and explained.

2. RELATED WORK

Several models have been used over the years to quantify CLV. They have ranged from simplistic to complex, and they have incorporated ideas from diverse fields such as mathematics, econometrics, and computer science. The Multitask learning model makes use of some of these models, and aims to improve on all of them when data is sparse in some areas but rich in others.

2.1 RFM Model

One of the first attempts to gauge customer value was a system called the RFM model, standing for Recency, Frequency, and Monetary Value. These models were originally intended to determine how successful a direct marketing campaign (e.g. postcards) would be to each customer, individually. They incorporated only three variables: the time between now and the customer’s last purchase, the average time between a customer’s purchases, and how much money the customer spends on any given purchase (Wei et al. [19]). The most typical application of this model would break customers up by quintiles on each of the three factors being looked at, yielding 125 groups of customers. Based on some formula, the costs of marketing to each cell and the predicted value from each cell would be calculated, and a breakeven line would be formed. The company would then deploy its marketing campaign to only target those cells which it considers could be profitable.

This technique can be fairly readily applied so as to determine CLV. The technique is simple, intuitive, and does not require a large amount of complicated data. Essentially, a company wishing to determine CLV of its customers would assume a constant direct marketing campaign for the remainder of each customer’s lifetime, and then determine the profitability of each customer (Cheng et al. [3]). However, this technique has a number of drawbacks. The concept of a constant marketing campaign is impractical; the campaign itself would be incredibly expensive and the customers would eventually grow immune to it. Looking only at transaction histories without accounting for demographic factors could lead to major oversights. Unless the formula is manually tweaked to backtest better, there is no impetus for the model to learn from past successes and failures.

The proposed approach will improve on the basic RFM model in a variety of ways. Most prominently, the technique incorporates learning, thereby increasing the strength of the model over time. It also accounts for sequential data, rather than selecting a fixed time period and looking only at that bucket of data. Finally, as has been shown elsewhere (Gupta et al. [8]), models which incorporate more than these three factors do a better job predicting CLV than RFM models. Although RFM was not developed explicitly to calculate CLV, it is an important benchmark to be compared against.

2.2 Econometric Models

Some models exist that seek to take more covariates into account than probability models, which typically only use recency, frequency and monetary value of transactions in order to estimate the different elements of CLV.

An example of this is the use of proportional hazard models to estimate how long customers will remain with the firm. The general form of these equations is

\[ \lambda(t, X) = \lambda_0(t) \exp(\beta X) \]

where \( \lambda \) is the hazard function, which is the probability that the customer will leave the firm at time \( t \) given that they have remained with the firm until time \( t \), and \( X \) is a set of covariates which may depend on time. \( \lambda_0 \) is the base hazard rate; typical examples include the exponential model (in which case it is memoryless) or the Weibull model (to capture time dependence). See Cox [4] for the original paper on proportional hazard models, and Knott et al. [11] for an example of use.

Once this hazard rate has been established, the survivor function is calculated as

\[ S(t) = P(T \geq t) = \exp \left( -\int_0^t \lambda(u) \, du \right) \]

Where \( T \) is the actual time that the customer leaves the firm. For the exponential distribution, the hazard function is constant; this makes estimating the survival function relatively simple. There is also the following:

\[ P(T < t) = 1 - S(t) \]

If the time that customers leave the firm is known, the likelihood of customers leaving can be estimated. Hence, to predict the function parameters (\( \beta \)), the total likelihood can be maximized (or minimize log likelihood).

While this use of covariates take more into consideration than probability models, it focuses on aspects of CLV independently (for example, treating customer retention and the amount that customers spend separately). The proposed approach treats CLV as a final goal, allowing the shared basis, a representation of basis vectors to define each of the tasks, to act across these components.

Furthermore, these models treat all customers equally. The proposed approach will be able to extract more from the given data by dividing the customers into sections, while still allowing each segment to learn from the others via the shared basis.

2.3 Persistence Model

Persistence models focus on modeling the behavior of components such as acquisition, retention and cross selling. These
models can be used to study how a change in one variable (such as a customer acquisition campaign) impacts other system variables over time. This approach can be used to study the impact of advertising, discounting and product quality on customer equity. It projects the long run or equilibrium behavior of a variable or group of variables of interest.

For example, a firm’s acquisition campaign may be successful and bring in new customers (consumer response). That success may prompt the firm to invest in additional campaigns (performance feedback) and possibly finance these campaigns by diverting funds from other parts of its marketing mix (decision rules). At the same time, the firm’s competitors, fearful of a decline in market share, may counter with their own acquisition campaigns (competitive reaction). Depending on the relative strength of these influence mechanisms, a long-run outcome will emerge that may or may not be favorable to the initiating firm. Dynamic systems can be developed to study the long run impact of a variable and its relationship with other variables.

Persistence models are well suited for CLV because it is a long term performance metric. Such models help quantify the relative importance of the various influence mechanisms in long-term customer value development, including customer selection, method of acquisition, word of mouth generation, and competitive reaction. However, this approach cannot perform well when there is little data to work with, as it relies on consumer behaviors that only reveal themselves over longer periods of time.

2.4 Diffusion Model

Unlike the other approaches, the prime focus of the diffusion model is on Customer Equity (CE). The purpose of the more aggregated approach is that CLV often restricts itself to focus on customer selection, customer segmentation, campaign management, and customer targeting. A broader approach of integrating the CLV of both current and future customers can help produce a strategic metric useful for higher-level executives.

In essence, CE is the sum of the CLV of future and current customers. There are two key approaches used in the measurement of CE. The first is the production of probability models of acquiring a certain consumer using disaggregate data (Thomas 2001 [17]; Thomas, Blattberg, and Fox 2004 [18]). This is primarily the methodology used by the approaches described so far. The alternate approach is to use aggregate data and diffusion/growth models to predict number of customers likely to be acquired in the future.

Gupta, Lehmann, and Stuart [9] showed that CE is a good estimate for 60% of the five companies investigated. The exceptions included eBay and Amazon, an indication of the weaknesses for the model to be applied with larger online retail firms. A particularly interesting insight of this approach is the relative importance of marketing and financial instruments, where a 1% change in retention would negatively affect the CE by 5%. This is in stark contrast with a similar change in discount rate producing only a 0.9% change in CE. For example, a $45 million expenditure by Puffs facial tissues to increase its ads awareness ratings by 0.3 points will compound to eventually result in producing a $58.1 million in CE.

Diffusion models can also be used to assess the value of a lost customer (Hogan, Lemon, Libai [10]). The key premise of this proposition is that firms that lose a consumer would not only lose the customer’s CLV, but also the word-of-mouth effect generated by that consumer. This indirect effect can be compounded to a four-fold effect, as investigated in the online banking industry. The above proposition of incorporating lost consumers in the model is of key importance in prediction of lifelong customer value, due to the triggering of a butterfly effect on the strength of the customer base.

2.5 Probability Models

Researching existing techniques for calculating CLV shows that the Pareto/NBD (Schmittlein, Morrison and Colombo [16]) is one of the most widely used methods. The proposed model applies the Pareto/NBD to each separate product category. In Fader et al. [7], a paper that implements the Pareto/NBD model, CLV is defined as follows:

\[ \text{CLV} = \text{margin} \times \text{transaction revenue} \times \text{DET} \]

*Margin:* How much profit the firm makes per sale to this customer. This data is typically unavailable to researchers (for big companies, financial statements may be useful), but does not really affect trends in data. It is a value between 0 and 1.

*Transaction Value:* How much the customer is expected to spend during each transaction. Multiplied together with margin, this gives the monetary value of each transaction to the firm.

*DET - Discounted Expected Transactions:* This is the total number of transactions a customer is expected to make in her lifetime. Transactions that are further in the future are discounted in order to take the time value of money into account.

For model fitting purposes, it may not be realistic to calculate lifetime transactions; this is a difficult number to find in order to test against (since customers may stay with a firm for many years). A cleaner approach for model fitting is to choose a certain time horizon and to estimate the number of transactions that a customer will do in the period; this can then be compared to actual data for training. Similarly, if margin information is unavailable, it would be better to leave it from calculations while training models. Therefore, a new metric, “Future Customer Spend”, is defined to be tested against. Note that once models have been estimated, the parameters can be used to build full CLV estimations again.

\[ \text{FCS}(t) = \text{trans}(t) \times \text{transaction revenue} \]

Where trans(t) is the expected number of transactions in the next t periods.

Two factors need to be estimated to calculate FCS: the number of future transactions a customer is likely to make, and expected revenue per transaction. Following Fader et al. [7], these factors are assumed to be independent and are modeled separately, using the Pareto/NBD and Gamma/Gamma models.

2.5.1 Pareto/NBD

The Pareto/NBD model is used to predict the future transactions a customer will make, and relies on the following assumptions:

- Customers are “alive” for a certain amount of time, after which they are permanently inactive.
3. SYSTEM MODEL

Current industry standards follow a relatively simple approach to CLV estimation, since most firms take the product of past transaction values and the forecast period in the future to provide a numerical value of CLV. More refined models use the Pareto/NBD and the Gamma/Gamma in order to predict future CLV, as shown in the Related Works section.

First, a brief overview of the proposed model is provided. Second, the current existing standards of probability models are examined and their shortcomings are noted. Third, the proposed model is expounded upon to provide to delve into more details. Finally, some key applications of the model are listed.

The proposed system extends current naive and advanced industry standards by taking the model forward by incorporating multitask learning. When the solutions to certain tasks are related through some underlying structure, an agent may use multitask learning to share knowledge between tasks. In the proposed model, first, the transactions are segmented by product category to produce an individual model for each of the categories. Second, an individual model is produced for each of the product categories, outputting parameters for the Pareto/NBD and the Gamma/Gamma distribution. Third, sparse coding is performed on the shared basis, which includes all parameters for each of the product categories, refining the parameters of the individual categories further. Finally, the refined parameters obtained from the multitask approach are then used to calculate the expected value of CLV for a certain time span.

Figure 1 explains how the model splits products by product categories and applies a Pareto/NBD and Gamma/Gamma model to each category individually. In Figure 1, samples drawn from product categories 1, 2, and 3 are sufficient to fit the statistical model individually to each of those categories. For simplicity of the figure, \( \alpha \) and \( \beta \) are used to represent model parameters rather than the full seven parameters that will be used in reality. For each model that fitted, parameter vectors \( \alpha \) and \( \beta \) describe the product category. In the figure, \( \alpha_1 \) and \( \beta_1 \) are the model parameters for category 1, \( \alpha_2 \) and \( \beta_2 \) are the parameters for category 2, while \( \alpha_3 \) and \( \beta_3 \) are parameters for category 3.

Consider a new product category, product category 4, with sparse data. Data is considered to be sparse if it has a training set of less than 150 data points to produce a well-fitted model. The model encounters the statistical problem that a small sample may not capture the full range of the population. Any model estimated for the above category would be subject to overfitting, primarily due to the lack of data points for the generation of an accurate model. This can have drastically negative effects, including misallocation of resources and ineffective strategy and planning. This problem may arise either because product category 4 is new, or because of inherent changes in the market structure yielding less data.

At this point, the single model approach would be to simply fit a model across all the product categories and then apply that model’s parameters to describe category 4 (along with all other categories). The separate model approach aims to avoid discarding granularity; the implementation of Multitask learning is used to solve this problem instead (as illustrated in Figures 2 and 3).

Multitask learning enables knowledge acquisition for re-
In the paradigm of multitask learning, several related tasks are estimated using probability models, a shared basis is produced. The shared basis is a set of linearly independent vectors, which in a linear combination, can represent the entire vector set. This shared basis includes rows of the seven parameters estimated, four from the Pareto/NBD and three from Gamma/Gamma, for each of the product categories.

In the paradigm of multitask learning, several related tasks of CLV estimation are learned together by sharing knowledge between the tasks. It is important to note that the tasks in the shared basis can selectively share information with other tasks, resulting in a more independent and dynamic system. Consider the representation of each task parameter vector to be a linear combination of a finite number of underlying basis tasks. The coefficients of the linear combination are generally sparse for the tasks being estimated. Moreover, there is an overlap in the sparsity patterns between different tasks, which can determine the degree of sharing in the model. Since each task is only represented by a vector of seven parameters, the model exists in a low-dimensional subspace allowing the tasks to overlap with each other in multiple bases. This form of Machine Learning is known as sparse coding, a technique used to represent an input signal by a small number of basis functions [13]. Sparse coding outputs a basis set and a weight vector, which, when combined, reconstructs the signal to integrate shared knowledge from other tasks. This product of the basis matrix and the weight vector hence produces a new matrix of the refined parameters for each of the product categories.

Consider a finite but large set of transaction data in the training set $X = [x_1, x_2, x_3, \ldots x_n]$ in $\mathbb{R}^{m \times n}$. Here, $n$ is the number product categories being modelled, and $m$ is the seven parameters described above. Let $D$ in $\mathbb{R}^{m \times k}$ be the dictionary representation, wherein each column represents a basis vector. Hence, $X$ admits a sparse approximation over the dictionary with $a$ atoms. Let $L(x, D)$ be a loss function which must be minimized for a well fitted $D$. The number of product categories, $n$, is larger than the seven model parameters implemented, as is typical for applications of this type of learning. Moreover, the degree of sharing, $k$, is at most equal to $n$, since sparse coding is implemented rather than an overcomplete dictionary. This degree of sharing is a control knob that can be moved along the spectrum to find the optimal solution. The loss function $L(x, D)$ is the optimal value of the Lasso regularization problem with the regularization parameter $\lambda$:

$$L(x, D) = \min_{\alpha \in \mathbb{R}^k} \frac{1}{2} \| x - Da \|^2 + \lambda \| \alpha \|_1.$$  

Here, $\lambda$ is a regularization parameter. A fundamental characteristic of this technique is that this $l_1$ penalty results in a sparse solution for $\alpha$ in the basis pursuit. Sparse coding hence enables a representation of shared basis vectors, which further refine the individual parameters.

One of the challenges of sparse coding is the determination of the optimal degree of sharing. The degree of sharing may lie in the spectrum between completely individualized models to a single model with all data being shared. In order to determine the best degree of sharing, the model uses an out-of-sample testing procedure to compare the CLV estimates produced by the shared representation with the actual CLV. The $\lambda$ producing the model with the least percentage error is considered optimal. Hence, the development set is leveraged to ensure that an overfitted model is not chosen. These parameters are then used to estimate the conditional expected value of the amount a customer will spend, in order to determine the value of the CLV per category for each of the customers. For any customer, the aggregate of the CLV for all product categories is the projected future CLV.

A robust implementation of the system is provided, and experiments are run to compare the system with existing models. This enables testing the validity and reliability of the proposed model. Further details are provided in Section 4.
4. SYSTEM IMPLEMENTATION

In order to compare the proposed model to existing models, two different models are being generated using the data available. One uses the existing single model approach in industry, and the second applies the proposed solution (the shared knowledge model). In order to evaluate the strength of the proposed model, it was tested on data comprising 82,000 anonymized rows of transactions of a large online retailer, Amazon, from January 2012 to December 2012. The data (see Appendix A) contains rows of transaction data, consisting of several variables such as time of transaction, the relevant product category, item purchased, customer ID, zip code, average income level, etc. Note that since the data is anonymized, for CLV estimation, each customer is identified with their unique customer ID. The data was obtained from ComScore, through the Wharton Research Data Services (WRDS).

Before starting, transactions are divided into training, development, and testing sets, with the training set preceding the development and testing set chronologically. A two month span is used for each of the sets. As described in Section 3, the metric used to test results is Future Customer Spend (FCS). The training period will be used to forecast how much customers will spend in the testing period. The development set is used to determine the degree of sharing which would be optimal for the given dataset. This is then verified by observing actual spending behavior in the test period.

To measure the effectiveness of building separate models for each category, the following steps are conducted:

i. The Pareto/NBD and Gamma/Gamma distributions are used to estimate CLV per customer using all transactions observed in the training period for each category;

ii. These parameters are then fed into a matrix containing the 7 parameters defining all the product categories;

iii. Sparse coding with 100 iterations is performed on the matrix, with an increasing degree of sharing;

iv. The matrix containing the refined parameters obtained for each degree of sharing is then used to estimate the conditional expected value of the CLV on the development set. This process is repeated for each of the matrices produced in the training set for every single degree of sharing;

v. For each of the degrees of sharing, the estimated CLV produced by the refined parameter matrix is compared to the actual CLV of the customers. The optimal degree of sharing is decided to be the one with the least absolute percentage error in CLV.

vi. The parameter matrix of the optimal degree of sharing is then used to calculate CLV over the testing period, enabling a robust methodology of CLV estimation.

vii. For evaluation of the system, the absolute percentage error in CLV estimation by the system model is compared to that of existing industry standards.

The entire system is an end-to-end pipeline implemented in MATLAB. The pipeline coded takes as input the file containing the rows of transactions, and outputs the CLV of each customer. In order to perform Sparse coding, the open-source Sparse Modelling Software developed by the French Institute for Research in Computer Science and Automation is used, in particular the mexTrainDL library ([14]). This enables a generic and intuitive system, whereby users can focus on primarily catering to the market, rather than putting in arduous efforts in plugging discrete pieces together.

The end-to-end pipeline also enabled running of several experiments to determine if certain dependent variables have a statistically significant effect. A key experiment to test if the model performs better with sparse data was run by increasing the number of buckets the data was divided into. The number of buckets was modified to range between 5 and 8, in order to vary the sparsity of the training and development data used. Through tuning in the input parameters to the system, the entire package was made modular, leading to fewer errors in model design as well as a much more intuitive user experience.

Key insights are visualized through a consumer facing web application which plots the geographic distribution of the customer lifetime values across all product categories using the Google Maps API. Besides the Google Maps API, the application also makes use of the Bootstrap framework for stylistic elements, the D3 framework ([1]) for rendering bar and pie charts, and the Springy.js framework ([5]) for force directed distance graphs that are used to show relationships between different product categories. More details are provided in Section 6.
5. RESULTS

The effective system implementation described above leads to key results to test the validity and reliability of the proposed model. The overall results are obtained from comparing the shared model approach with the individual model and the industry standard (overall model). There are two prime findings of this model that extends industry standards: (i) Multitask learning produces much more robust results in product categories with sparse data; (ii) Sharing knowledge decreases the variance in error observed.

In order to evaluate the estimation error for the shared and individual model approaches, the absolute percentage error for each product category is plotted against the number of customers in the each product category. As can be seen in Figure 5, for product categories with less than a 150 customers who made transactions, the shared model approach brings the absolute error down to as low as 0.6 compared to the 0.9 error obtained by the individual model. Therefore, this shows that in order to model product categories with data points fewer than 150, the shared model approach should be used. This also proposes that a combination model would work best for the entire data set: Use the shared model approach to model product categories with fewer than 150 customers who have made transactions, and use the individual model approach for product categories with more than a 150 such customers.

Figure 5: The shared model approach improves estimation of CLV for small product categories.

Other than the absolute error per category, it is also interesting to measure the mean absolute error across categories. The fraction of customers taken into account in the model is varied, and the mean absolute error is calculated for both the shared model and the industry standard model approaches. For different fractions of customers, the average error is similar between the proposed model and the industry standard. According to this measure of error, the proposed model is on track with current implementations, but presents better recall than existing models.

Figure 6 indicates the difference between the standard deviation of error for the different fractions of customers taken into account in the models. Much lower standard deviation is found in the shared model as compared to the industry standard approach. This gives a strong signal of indication that CLV can be calculated more consistently across categories in the shared model approach, and lead to an overall better product category fit. This is likely due to the fact that a model built using all categories heavily favors large categories, leaving small categories prone to high degrees of error. Once again, this may suggest a hybrid-model approach for an overarching solution as described above.

Figure 6: The shared model approach leads to lower variation in error estimations.

A final study was done to test the stability of results. For comparison purposes, the degree of sharing $k$ was fixed at 5. Then, three models were trained on four months of data, and tested on two months of subsequent data in order to find the error in estimation.

<table>
<thead>
<tr>
<th>Seg. 1</th>
<th>Seg. 2</th>
<th>Seg. 3</th>
<th>Seg. 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Size</td>
<td>105.34</td>
<td>274.43</td>
<td>411.00</td>
</tr>
<tr>
<td>Categories</td>
<td>116</td>
<td>40</td>
<td>6</td>
</tr>
<tr>
<td>Individual Model</td>
<td>4.38</td>
<td>0.52</td>
<td>0.36</td>
</tr>
<tr>
<td>Overall Model</td>
<td>3.89</td>
<td>1.27</td>
<td>0.52</td>
</tr>
<tr>
<td>Shared Model</td>
<td>1.68</td>
<td>0.50</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Table 1: Mean error rates for the different models are shown. The shared model performs best with small product categories.

This process was repeated five times with five time periods, starting on January 1, January 16, February 1, February 15, and March 1. All categories were then pooled and divided into the four segments as shown in Table 1. Segment 1 contains all categories with fewer than 200 customers making transactions in the training period; segment 2 contains
all categories with more than 200 and fewer than 400 such customers; and segments 3 and 4 follow the pattern (400-600 and 600-800 such customers respectively).

The results of this study is shown in Table 1. Even repeated across a larger sample, it still holds true that the shared model outperforms the other models for small product categories, while remaining competitive for larger product categories.

6. FINAL PRODUCT

The final product, CustoVal, is a consumer facing front-end web application that uses JavaScript, HTML, CSS, PHP and Python technologies. Once the output from the CLV analysis is obtained, it is transferred via a python script to the cloud and stored in a format that can be directly read from the application.

The application focuses on providing visualizations and insights to marketing professionals to enable informed decision-making on how much to spend on targeting customers from a specific product category and geographic location, as well as how to maximize a customers spend across all product categories. There are two modes the application can exist in: (i) The overall insight mode, highlighting key insights for all product categories, and (ii) The per category mode, enabling the user to pick a product category and view insights specific to that category.

The overall view is shown in Figure 7. The Google Maps API effectively highlights the correlation between geographic location and CLV, by comprehensively depicting the geographic layout of future customer transactions. The customers are represented as circles on the US map in the location where they made their purchase. The size of the circles represents the transaction value and overlapping transactions result in circles of higher color intensities indicating higher frequencies of transactions. In addition to geographical distribution, the “all category” view also provides three pie charts that show the share of transactions made in each of the product categories for all customers, the top 10% of the customers, and the top 5% of the customers. The pie charts for the top 5

The per category view, displayed in Figure 8, also provides a geographical visualization but only of the customers that made purchases in that product category. In addition, this view provides the customer acquisition cost for that product category, and also a force graph showcasing the relationship of the product category with others. The spring graph is developed by calculating the mean distance of the product category parameter from the parameters of other product categories. This visualization is meant to provide marketing professionals with insights as to which product categories are similar to which others and how they can cross sell to customers across product categories.

Along with the geographic rendering, several other tools such as the Bootstrap and D3 framework ([1]) are used for rendering the graphs and visualizing the data in an intuitive manner. Moreover, the Springy.js framework ([5]) is used to depict force directed distance graphs, indicating the correlation between different product categories observed in the data.

These are the features currently available in Version 1 of CustoVal allowing firms to enable research in Machine Learning to make strategic decisions regarding customer acquisition and targeting. There are several opportunities for the types of applications the proposed model could have on the industry, especially in enhancing marketing strategies and reducing customer acquisition costs. Some of the most important applications are enumerated below:

1. Recover important customer spending potential across different product categories, which allows for more specific targeting of high value customers for a product category through marketing campaigns.

2. Improve retargeting and ROI (Return On Investment) for customers by showing them ads in other product categories of value to them.

3. Accurately predict customer spending potential for new product categories that have not seen many transactions yet.

On a concluding note, the key goal of this proposed model is to truly extend the current industry benchmark on estimating Customer Lifetime Value. These metrics can help firms leverage the data on customers, particularly in situations with sparse data, such as the launch of a new product category or a fundamental change in tastes or the market structure. Future iterations of the product will involve a more integrated pipeline that enables marketing managers to upload spreadsheets in excel format, which is used often in industry. Future work also involves providing more insights
of tastes, could have a spillover effect on the advertising they are subject to, as well as the products they purchase in the future. This can lead to a self-reinforcing feedback loop, which may result in tastes in a firm being extremely categorical and discrete. This can have larger social impact issues through stereotype and image propagation. This can be prevented by not taking extremely new and developing consumers into account in the development of the CLV model, as well as being careful in ensuring that the advertisement prevents reinforcements of negative portrayals and stereotypes.

8. FUTURE DEVELOPMENT

Models of customer purchasing behavior typically fall into four categories, defined by type of purchasing behavior and type of customer relationship. Purchasing behavior can be either discrete (e.g., a company with quarterly donation requests) or continuous (like most grocery stores: you can buy at any time). Customer relationships can be contractual (e.g., subscription-based services like cable television) or non-contractual.

CustoVal uses the Pareto/NBD, which applies to non-contractual continuous-time purchasing. The reasoning behind this was that most purchasing follows this behavior, but it is the most difficult to model. Hence, any improvement over current systems would have an impact. Future development of CustoVal would take the direction of exploring the other types of purchasing. The pipeline design allows for replacing only very specific parts of the system, which would make this process easier.

Furthermore, the system could be implemented as a hybrid model approach rather than strictly using the individual, overall, or shared models. It has been seen that the shared model does best on small product categories, while the larger product categories perform better with individual models. Thus, it would be worth investigating an implementation which chooses the optimal type of modelling to use based on product category size. Further research would need to be done in order to find the point at which to distinguish between small and large categories.

Moreover, the proposed system model weighed each of the categories equally for CLV estimation. This tended to work well since a sparse history would indicate a lower estimated CLV value. However, instead of using primarily history of transactions, the model could potentially extend itself to take characteristics of the consumer such as income, age, and gender to account. This could result in the product category models being weighed differently, with each of the weights learned from the characteristics of the consumers. This could incorporate an interesting element of customer features in analytics, providing more room for experimental research.

9. REFERENCES

[3] Ching-Hsue Cheng and You-Shyang Chen. Classifying the segmentation of customer value via rfm model and


### APPENDIX

#### A. DATA

From ComScore on the Wharton Research Data Services (WRDS) database, it is possible to obtain transaction data for major websites such as Amazon.com, with Machine ID (a unique identifier for each customer), transaction date and time, purchase amount and details, session details, demographic information, and zip code for each customer. The proposed model uses only the Machine ID, the number of transactions made by each customer, the date of the last transaction, and the average amount spent per transactions. Zip code is used in order to produce a geographic representation of the data.