Steve Rekuc

Final Report: Customer Lifetime Value Analysis

Problem Statement

Revenue growth is great to see in a young business, but sustaining that growth depends on how well a company retains customers as repeat buyers. Many companies can spend money on marketing to convince people to try their product; if those people don't like the product or the brand enough, it just requires more marketing money to acquire more customers in their place. A valuable company will be able to retain customers that try their products and therefore sustain their revenue growth.

We focus our modeling on a small online business that sells through shopify. The company began in 2013 selling skin care products and slowly grew their brand with only online sales. The company was acquired 2 years ago by a larger holding company that intended on boosting their sales before selling off the company again. We have those 7 years of sales data with 142,000 rows of purchased items and 72 rows of potential features to tease out any customer trends.

There are many different aspects of customer value or retention that we can investigate with this information; we look at the Shopify data as it relates to Customer Lifetime Value (CLV). Customer Lifetime Value is the total sales to a particular customer over their lifetime multiplied by the margin on those sales; this would give the total profit from sales to a customer. Since we don't know the margin on each of these products, we use total customer sales to each customer as the CLV. We focus on characteristics from a customer's first purchase that could indicate their CLV and we model that CLV from the features obtained in the first purchase. From this modeling, we hope the company can make better decisions to increase their CLV.

Data Wrangling and Anonymizing

We acquired shopify data from the holding company in the form of 5 separate CSV files that contained 142,000 rows and 72 columns. Before working with the data in a public course like this and showing it publicly on github, we had to remove any features that could be used to identify customers: email address, phone number, physical address. However, there are parts of this information that are useful as features and we also needed this information to help identify customers.

Shopify assigned customer numbers to the purchasers, but we found 63,085 unique customer numbers and 39,679 unique emails; this means that customers from the same email addresses were assigned multiple customer numbers. We used the customer emails to identify customers and widdle down those unique customer numbers to 39,679. We had to fill 585 missing customer ID's by using the row index; this gave us all unique customer ID numbers.

Before eliminating those identifying features, we extracted some potential features from those. While email addresses would be identifying, the email domain in their address would not be and could be a useful feature (some servers are more aggressive at blocking advertising emails). We extracted the email domain for the 25 most popular domains (even less common domains can be used to personally identify customers; rekucfamily.com or danscompany.com are too specific); we added these domains as a feature. We also extracted area code from the phone numbers in the hope that those would be useful identifiers; we then deleted the phone numbers. Missing area codes were filled with "other" since this is used as a categorical feature (even though it's a number).

We identified a few more features from the columns that would identify the company or the individual. The 'Vendor' column had the company's name or "Route"; we just changed this to 0 for our company and 1 for "Route" to preemptively create dummy variables. We changed the "Employee" column into a boolean to hide the employee names; null values in this column were filled with False, as there would be a value if that was an employee purchase (and the vast majority of purchases are not from employees). With those steps complete, the data was anonymized, so that one could not determine the company or customers from the data. We could then move on to looking at the other columns

Data Cleaning

With the data anonymized, we could start looking at the other columns for value. We immediately identified a bunch of columns that are not useful for this data set: 11 different columns of "Taxes"; "Notes"; "Note Attributes"; "Cancelled at"; "Fulfilled at"; "Receipt Number"; "Location"; "Device ID"; "Id"; "Risk Level"; "Currency" (everything was in USD); "Paid at"; "Payment Reference"; "Lineitem taxable"; "Lineitem fulfillment status". This cut us down to 30 columns of potential features in our dataframe. We then eliminated all of "Shopify Draft Orders" from the dataframe.

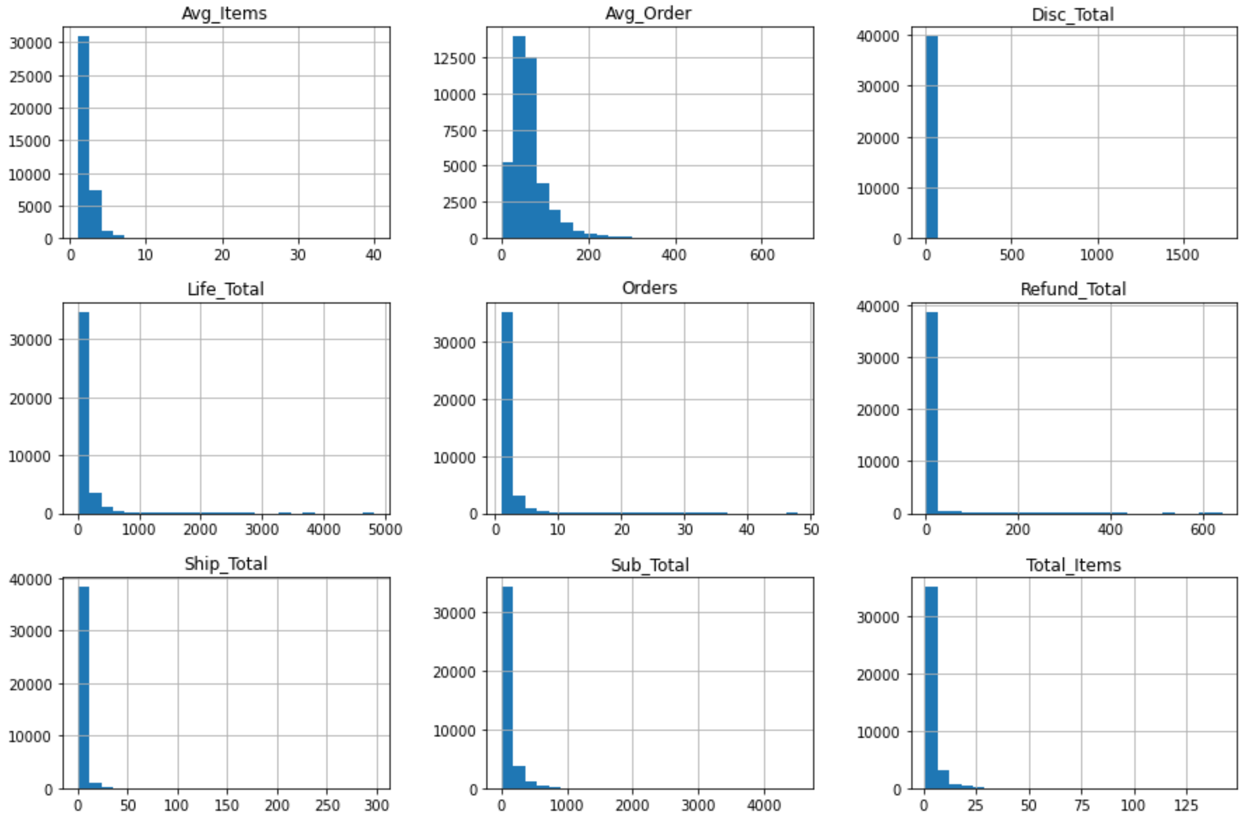
There were some missing values in our dataframe that needed to be filled, but most of those were categorical. In the case of "accepts marketing", we forward filled the values, as the first line of an order contains that information, but additional items in the order don't have that column filled. We also changed these values from Boolean to 1 for True and 0 for False to already create the dummy variable.

We continued in our process of filling missing values. We filled "Payment Method" with "unknown" for those missing values. For missing "Lineitem sku", we filled this with the "Lineitem name"; we could have potentially used a look-up between "Lineitem name" and "Lineitem sku" to see if other rows in the dataframe had the same description and could be used to find the correct SKU; since there was only 2050 values missing and we don't know if there would be an SKU corresponding to each of the names, we skipped this to save time. We filled the shipping method with "unknown" for those missing values.

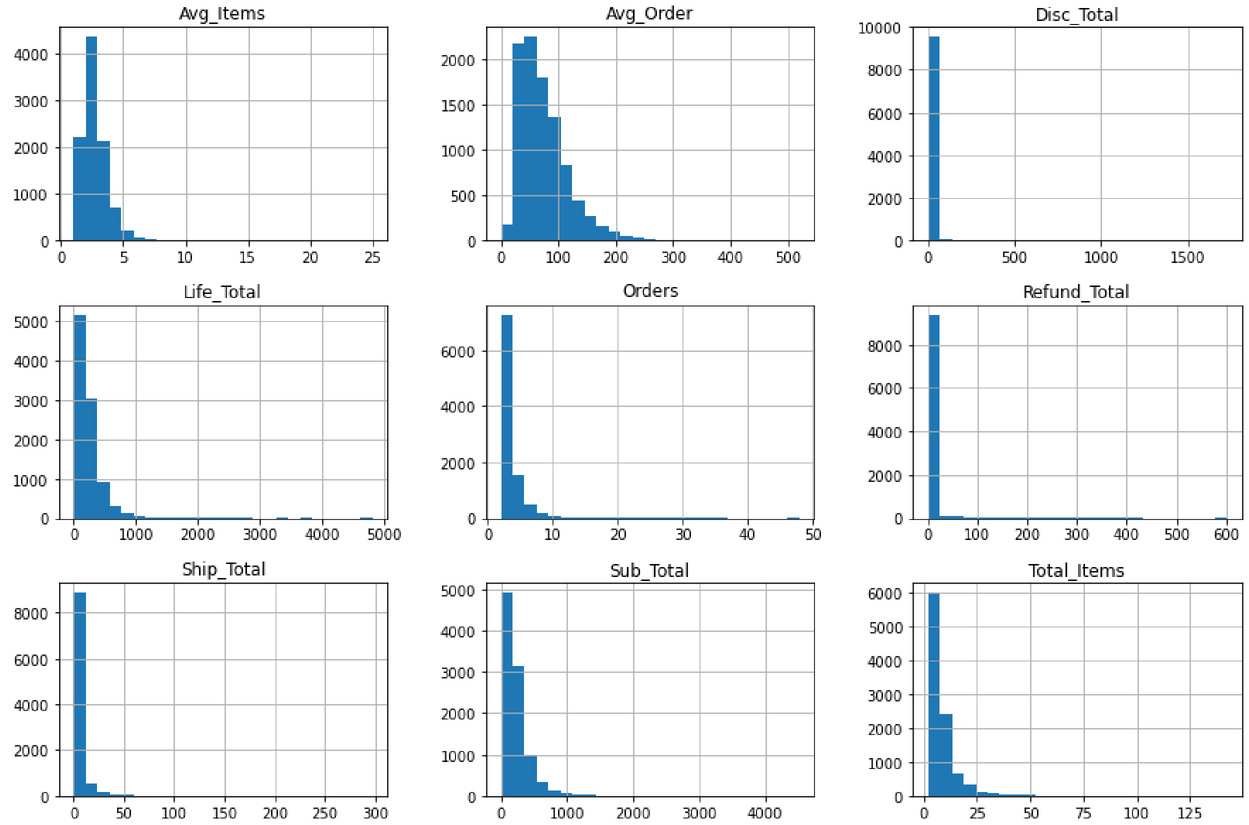
We created 3 separate dataframes for analysis: Items, Orders, and Customers. The items dataframe contained all of the items in an order; the Orders dataframe was aggregate to just each independent order with the lead SKU (no additional SKU's); the Customers dataframe was aggregated over the customers with and only contained value of their first order and the lead SKU on that order. By aggregating these down to smaller dataframes, it allowed us to just look at the features we needed to evaluate customer lifetime value. The Orders dataframe allowed us to look at some useful exploratory data analysis.

Exploratory Data Analysis

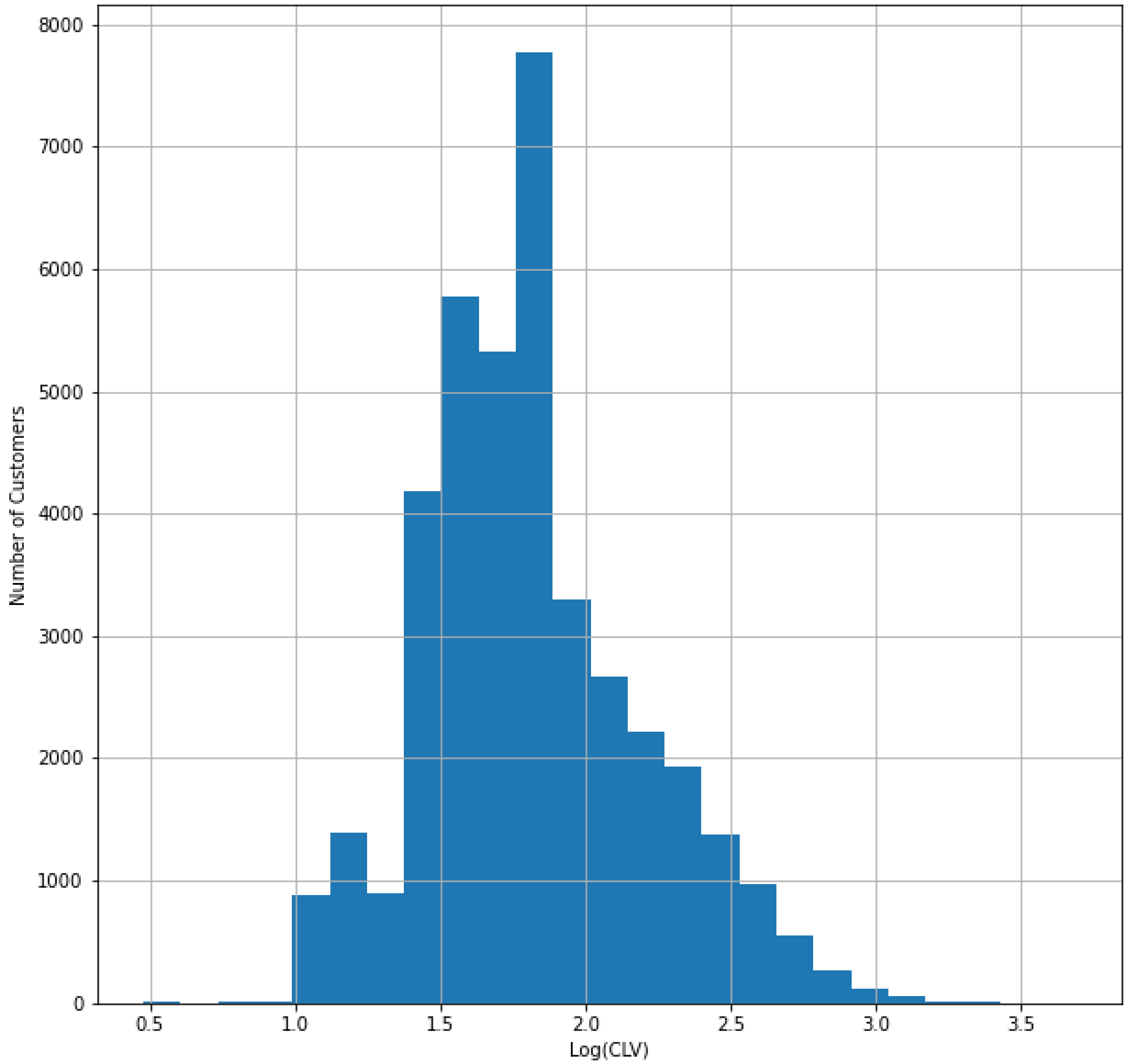
Now that we have our dataframes prepared, we investigated a lot of different aspects of the customer purchasing patterns. Let's first take a look at the distribution of the variables.



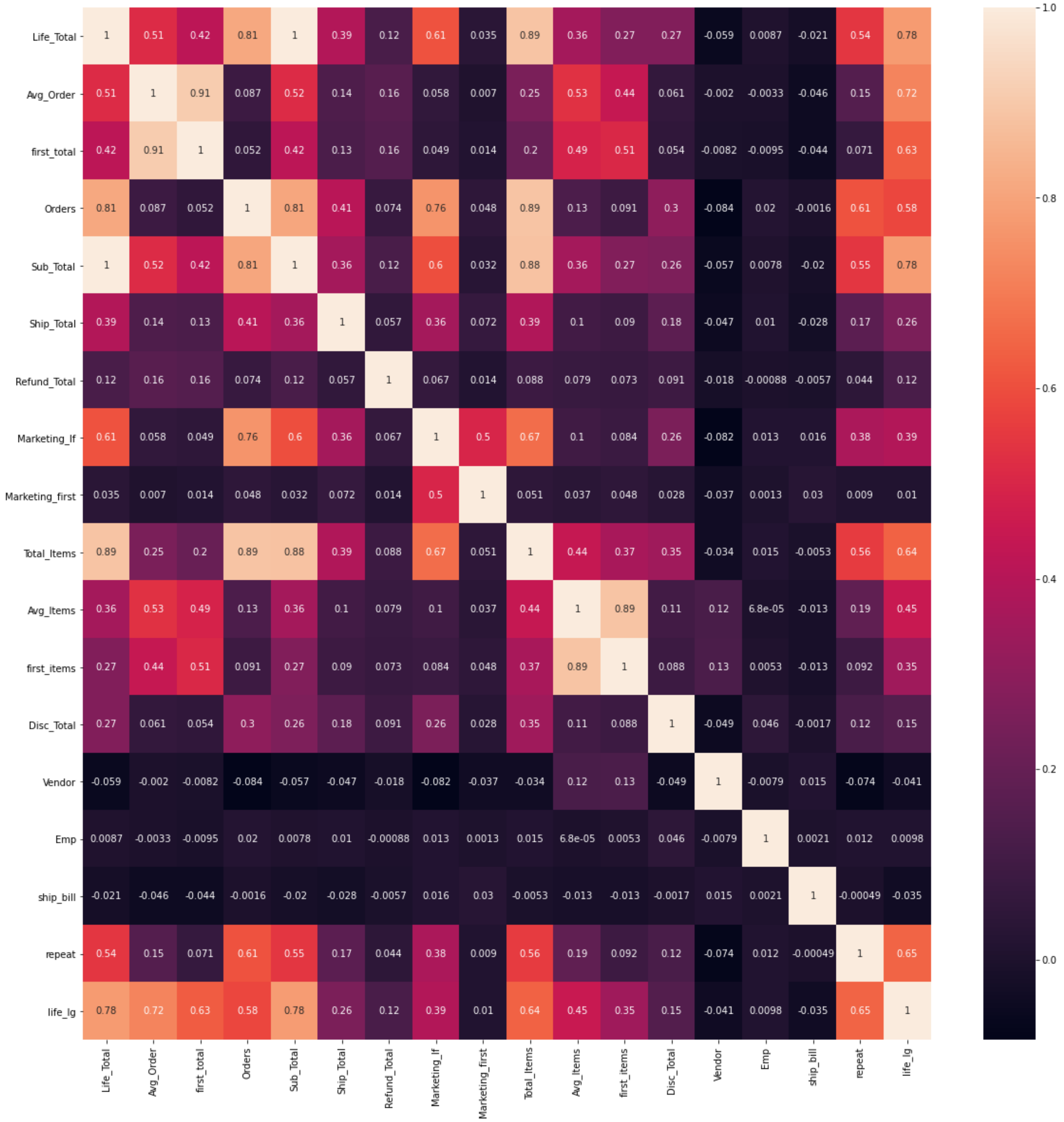
In the figure, we noticed that a lot of the variables have an exponential distribution: there are a lot of customers that purchase 1 order (or a couple of items), but fewer that come back for more. 24% (9699 out of 39771) of all customers placed a repeat order with one customer placing 48 orders on the high end. We looked at this distribution again - just considering the orders made by repeat customers (24% of customers).



The distributions are shifted slightly but still have that exponential (poisson) distribution. Because these values are exponentially distributed, we'll address this skew by taking the log of the Life_Total (Customer Lifetime Purchases) and use that for modeling. We considered taking the log of some more of these variables, as they are also exponentially distributed, but held off on that for now.



That looks a lot closer to a normal distribution; the skew went down from 6.99 for the CLV before transformation to 0.50 after taking the log values. We could also use the Box-Cox transformation to fix this skew, but the log value helped correct enough for now. The left side of the distribution is more jagged because items have discrete (not continuous) prices so it may be difficult to order something that has a log10 = 1.3 (~$20). We added this column to our dataframe and will be modeling based on this column because it's closer to a normal distribution Before we get any further into the modeling, let's look at the heatmap of the correlation coefficients.

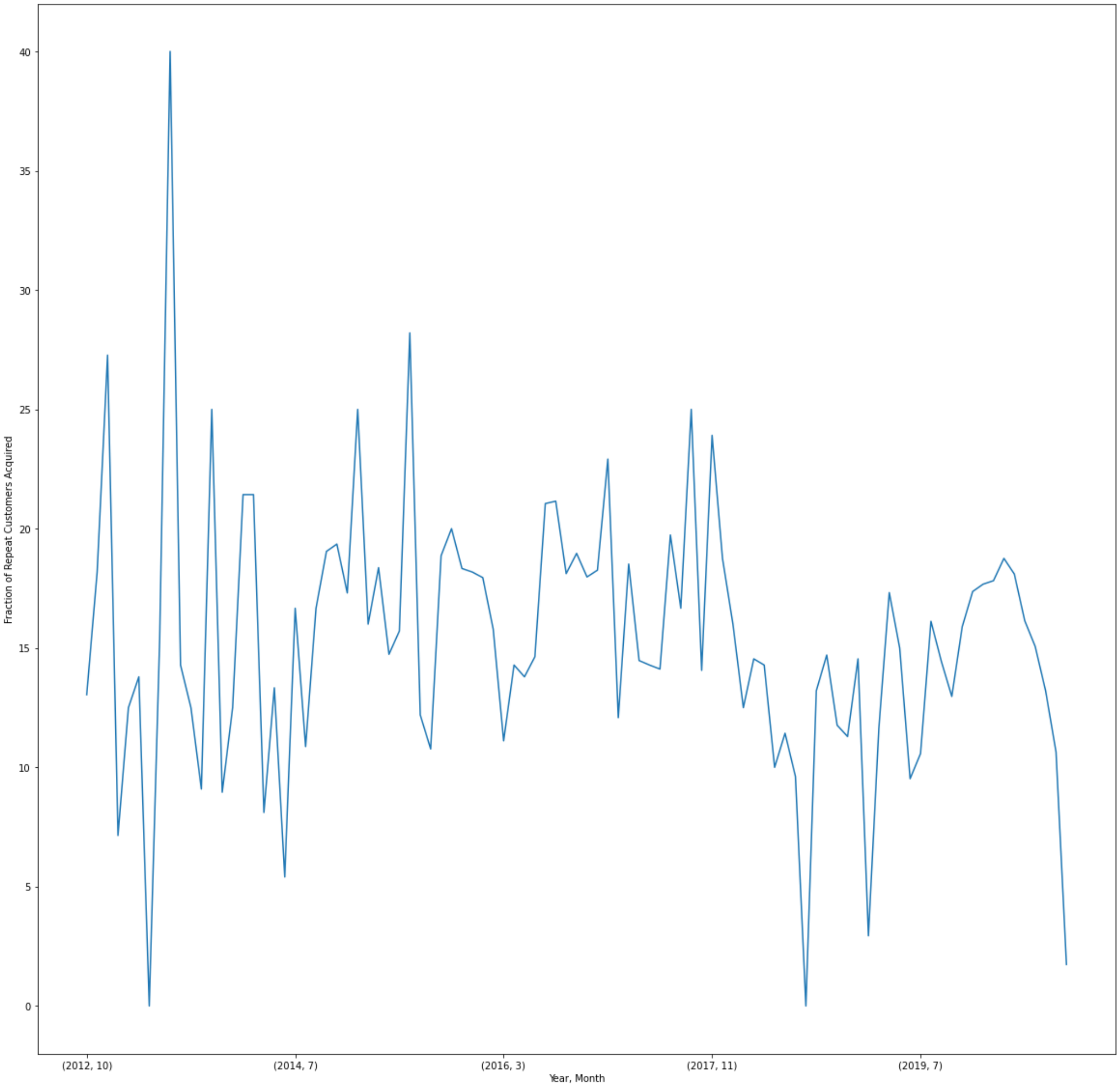


	Life_Total	Avg_Order	first_total	Orders	Sub_Total	Ship_Total	Refund_Total	Marketing_lf	Marketing_first	Total_Items	Avg_Items	first_items	Disc_Total	Vendor	Emp	ship_bill	repeat	life_lg
Life_Total	1	0.51	0.42	0.81	1	0.39	0.12	0.61	0.035	0.89	0.36	0.27	0.27	-0.059	0.0087	-0.021	0.54	0.78
Avg_Order	0.51	1	0.91	0.087	0.52	0.14	0.16	0.058	0.007	0.25	0.53	0.44	0.061	-0.002	-0.0033	-0.046	0.15	0.72
first_total	0.42	0.91	1	0.052	0.42	0.13	0.16	0.049	0.014	0.2	0.49	0.51	0.054	-0.0082	-0.0095	-0.044	0.071	0.63
Orders	0.81	0.087	0.052	1	0.81	0.41	0.074	0.76	0.048	0.89	0.13	0.091	0.3	-0.084	0.02	-0.0016	0.61	0.58
Sub_Total	1	0.52	0.42	0.81	1	0.36	0.12	0.6	0.032	0.88	0.36	0.27	0.26	-0.057	0.0078	-0.02	0.55	0.78
Ship_Total	0.39	0.14	0.13	0.41	0.36	1	0.057	0.36	0.072	0.39	0.1	0.09	0.18	-0.047	0.01	-0.028	0.17	0.26
Refund_Total	0.12	0.16	0.16	0.074	0.12	0.057	1	0.067	0.014	0.088	0.079	0.073	0.091	-0.018	-0.00088	-0.0057	0.044	0.12
Marketing_lf	0.61	0.058	0.049	0.76	0.6	0.36	0.067	1	0.5	0.67	0.1	0.084	0.26	-0.082	0.013	0.016	0.38	0.39
Marketing_first	0.035	0.007	0.014	0.048	0.032	0.072	0.014	0.5	1	0.051	0.037	0.048	0.028	-0.037	0.0013	0.03	0.009	0.01
Total_Items	0.89	0.25	0.2	0.89	0.88	0.39	0.088	0.67	0.051	1	0.44	0.37	0.35	-0.034	0.015	-0.0053	0.56	0.64
Avg_Items	0.36	0.53	0.49	0.13	0.36	0.1	0.079	0.1	0.037	0.44	1	0.89	0.11	0.12	6.8e-05	-0.013	0.19	0.45
first_items	0.27	0.44	0.51	0.091	0.27	0.09	0.073	0.084	0.048	0.37	0.89	1	0.088	0.13	0.0053	-0.013	0.092	0.35
Disc_Total	0.27	0.061	0.054	0.3	0.26	0.18	0.091	0.26	0.028	0.35	0.11	0.088	1	-0.049	0.046	-0.0017	0.12	0.15
Vendor	-0.059	-0.002	-0.0082	-0.084	-0.057	-0.047	-0.018	-0.082	-0.037	-0.034	0.12	0.13	-0.049	1	-0.0079	0.015	-0.074	-0.041
Emp	0.0087	-0.0033	-0.0095	0.02	0.0078	0.01	-0.00088	0.013	0.0013	0.015	6.8e-05	0.0053	0.046	-0.0079	1	0.0021	0.012	0.0098
ship_bill	-0.021	-0.046	-0.044	-0.0016	-0.02	-0.028	-0.0057	0.016	0.03	-0.0053	-0.013	-0.013	-0.0017	0.015	0.0021	1	-0.00049	-0.035
repeat	0.54	0.15	0.071	0.61	0.55	0.17	0.044	0.38	0.009	0.56	0.19	0.092	0.12	-0.074	0.012	-0.00049	1	0.65
life_lg	0.78	0.72	0.63	0.58	0.78	0.26	0.12	0.39	0.01	0.64	0.45	0.35	0.15	-0.041	0.0098	-0.035	0.65	1

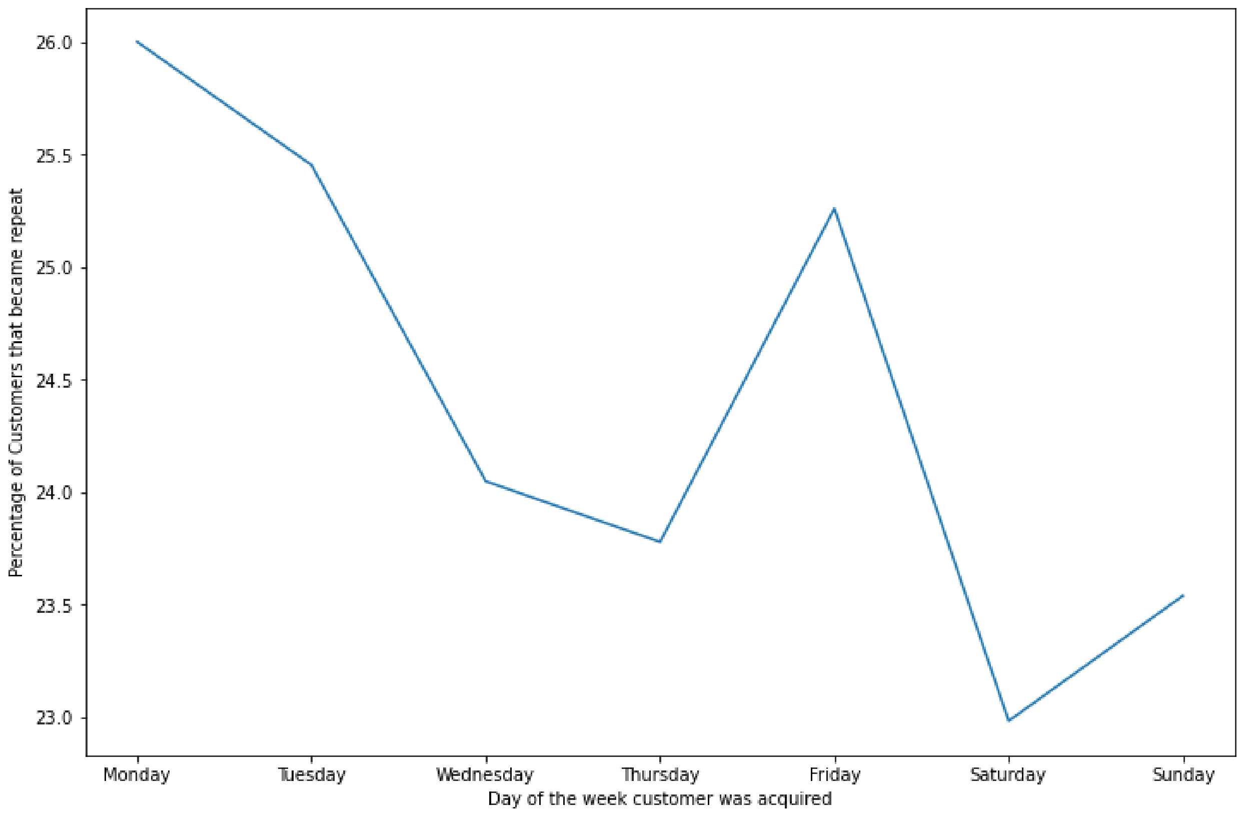
Since we included all of the columns in this heatmap, we see the obvious correlations between lifetime spending on number of orders, but there are also some important ones to notice: first order total and total lifetime (and log of total lifetime); number of items in the first order and lifetime total. Surprisingly, agreeing to marketing on the first order does not seem to have an impact on customer lifetime spending, but agreeing to marketing over the lifetime does have an impact on customer lifetime spending.

Based on this heatmap, we were also able to recognize some features that were not useful for use to keep, since they had very little correlation to CLV or the log of CLV: "Employee" and "ship_bill" were eliminated.

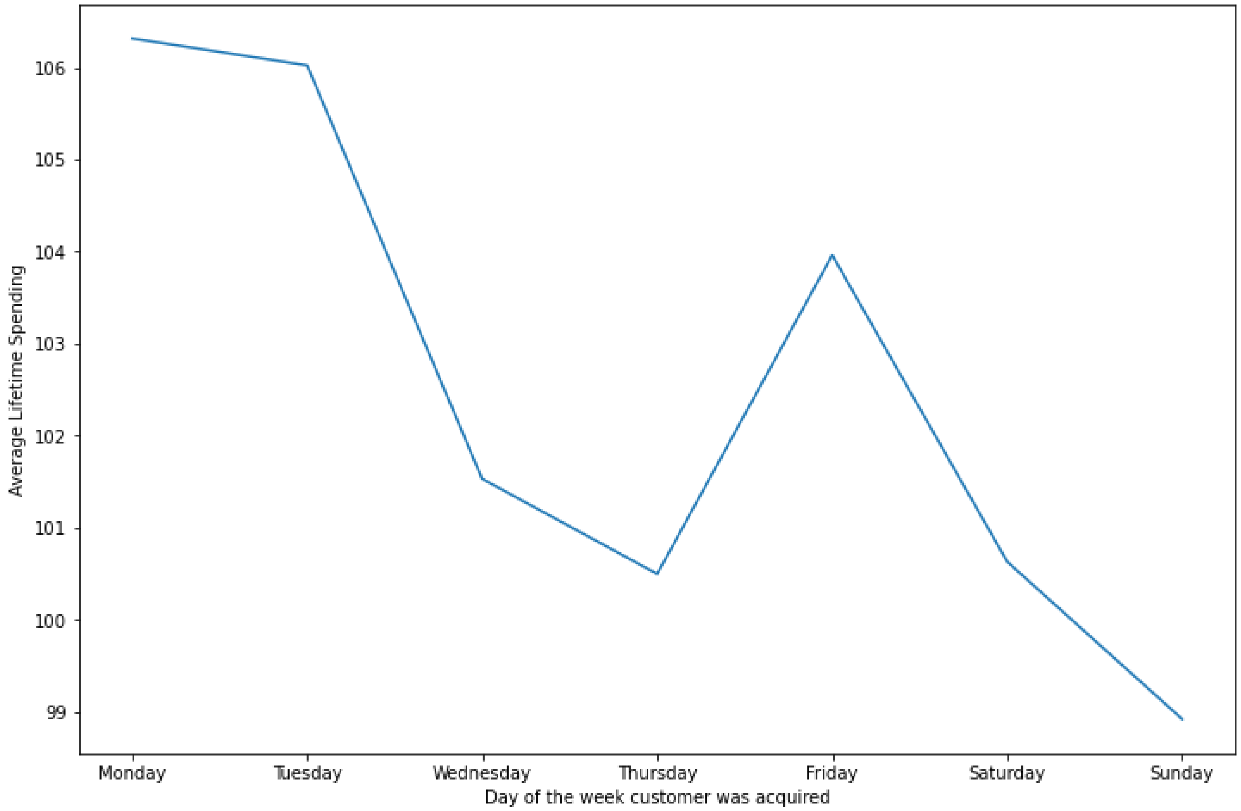
Before we move on to modeling, let's look to see if the repeat rate of customers differs at all based on when they placed their first order.



The repeat rate looks quite variable but hovering around that 20-25% over time. The dip in repeat customers that were acquired right at the end is probably because those customers didn't have enough time to repeat yet. We broke this down further and looked at repeat customers based on what day of the week they made their first purchase; let's look at that graph here.

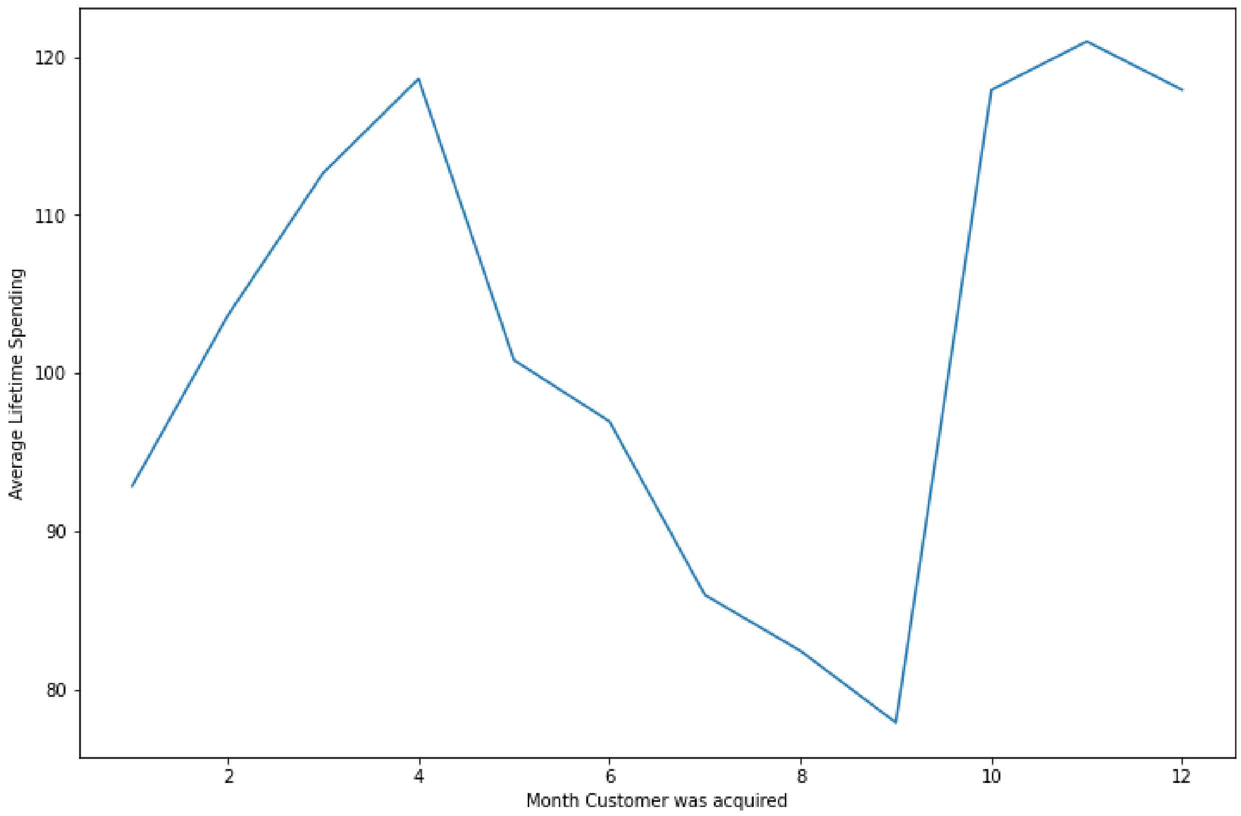


That looks like a small difference in repeat rate based on the day the customers were acquired; we considered the Customer Lifetime Value to see if that trend was also apparent when looking at spending.



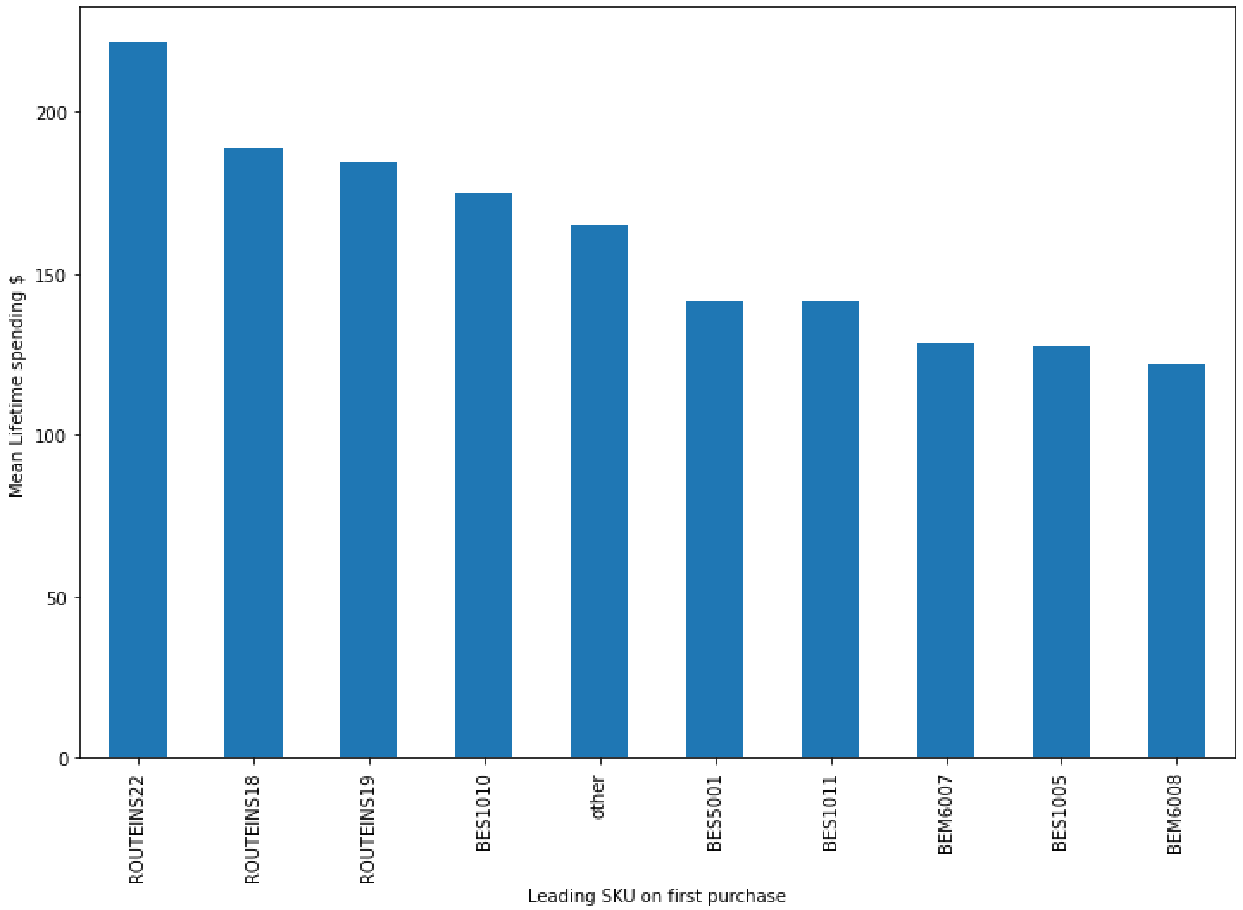
That looks like a significant difference between the CLV of customers acquired on Monday vs. Sunday. Since these look significant, we tested for Monday's average CLV being greater than Sunday's average CLV (null hypothesis of Monday mean CLV = Sunday mean CLV): z=2.715, p=0.0066. This means that there is a 99.3% confidence that Monday has a greater CLV average than Sunday. Tues and Friday look like pretty good days as well. While this is not exactly what we were modeling for, it provides good actionable insight.

We also looked at how the customer lifetime value varied depending on the month of the first purchase; we display those results in the figure below.



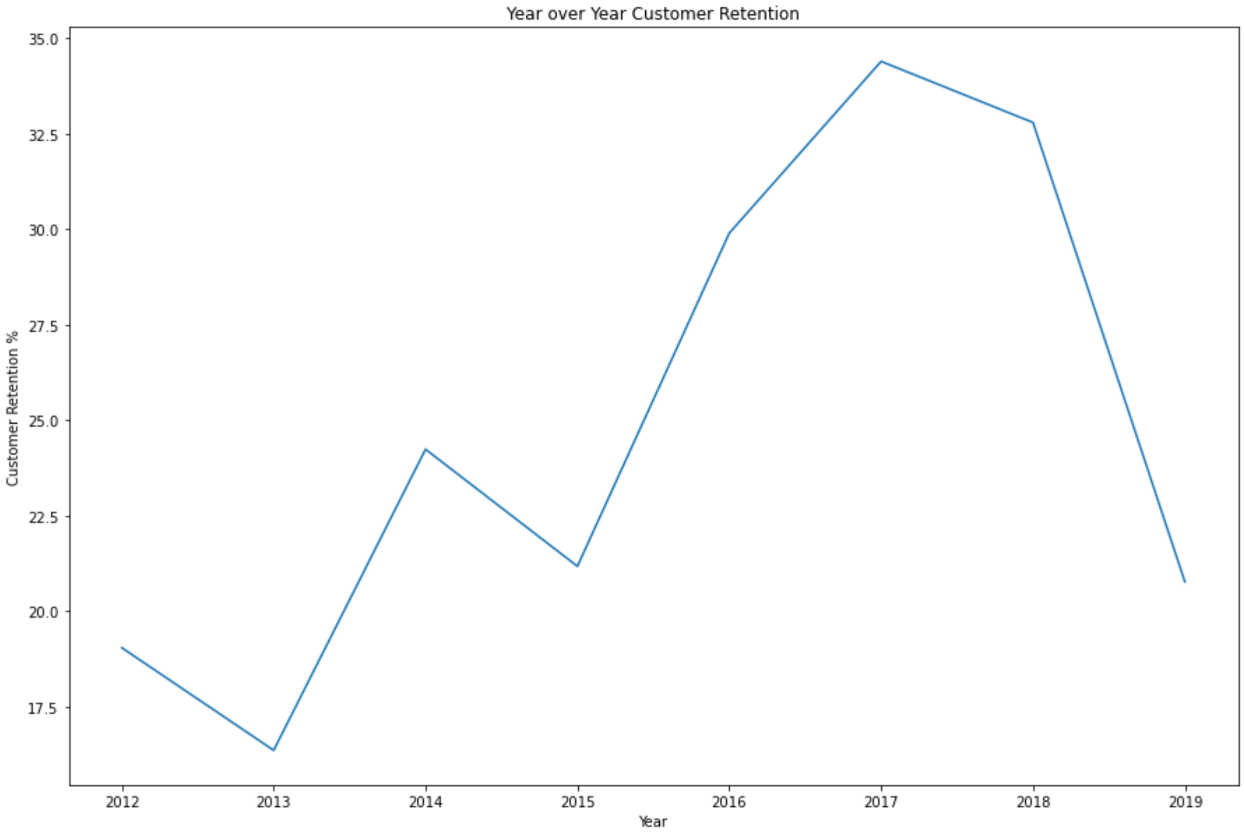
These results are even more drastic with higher valued customers acquired in October, November, December, and April. We tested that comparison between September and November (null hypothesis of September mean = November mean): z=6.89, p=6.05x10-12. The difference between those two values is unmistakable; customers acquired in November had much higher lifetime spending than those acquired in September. Since we drew this data from more limited months (only 7 of each), it's possible that some marketing in November (or lack of marketing in September) could account for the much better quality of customers acquired in November. We save that investigation for another time.

We also examined the effects of the lead SKU on the lifetime spending; we only considered the top 25 SKU's as some of the other lead SKU's occurred so infrequently that those instances are outliers. Belos is the plot from the top 10 performing SKU's from the top 25 most frequent lead SKU's.



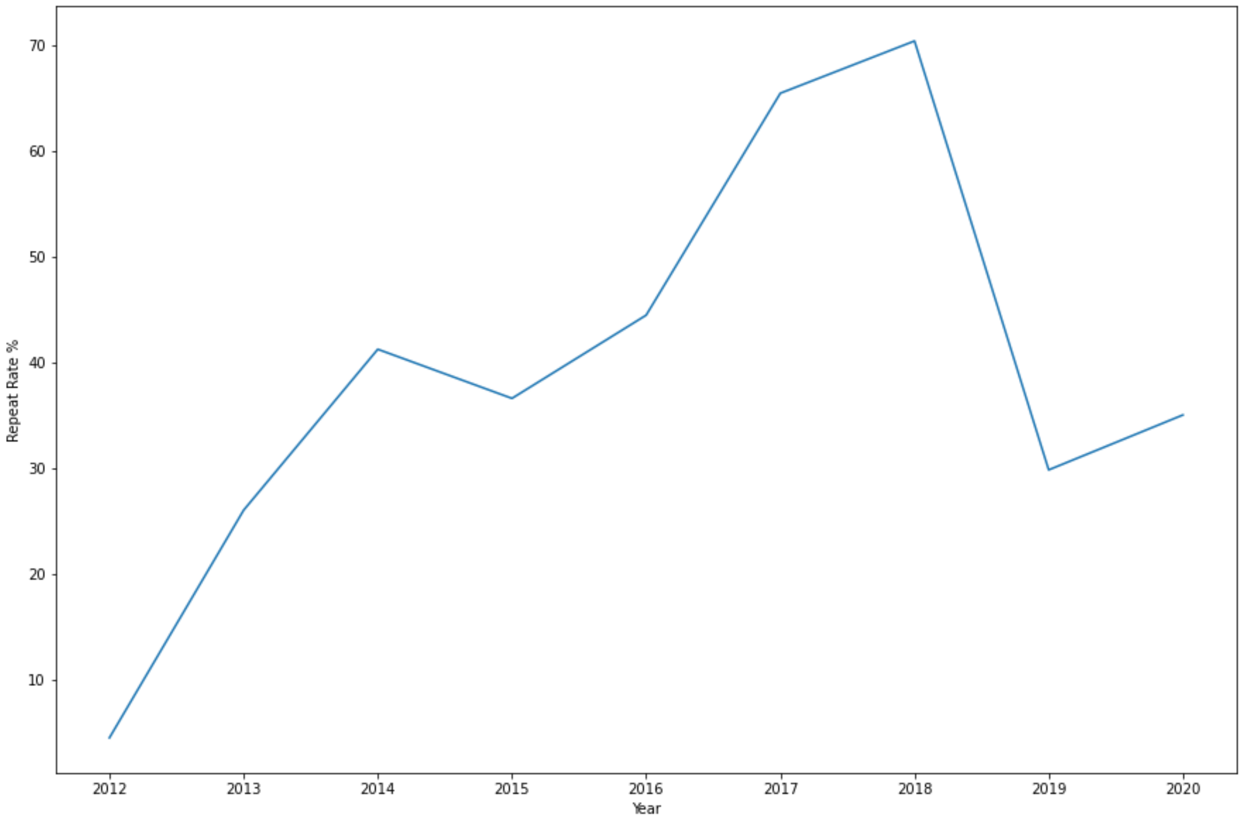
The mean lifetime customer spending is $102.44, so customers who purchased these SKU's as lead definitely continued to purchase more (over $210 in CLV for the top product). These are something we included as features in our model but did not investigate further in this report. We may spend more time diving deeper into this on future analysis.

In our analysis, we also took a quick look at retention rate and repeat rate. For the purpose of our analysis, we looked at retention rate as year over year purchases by a customer; therefore this is the percent of customers that purchase in one year and in the next year as well.

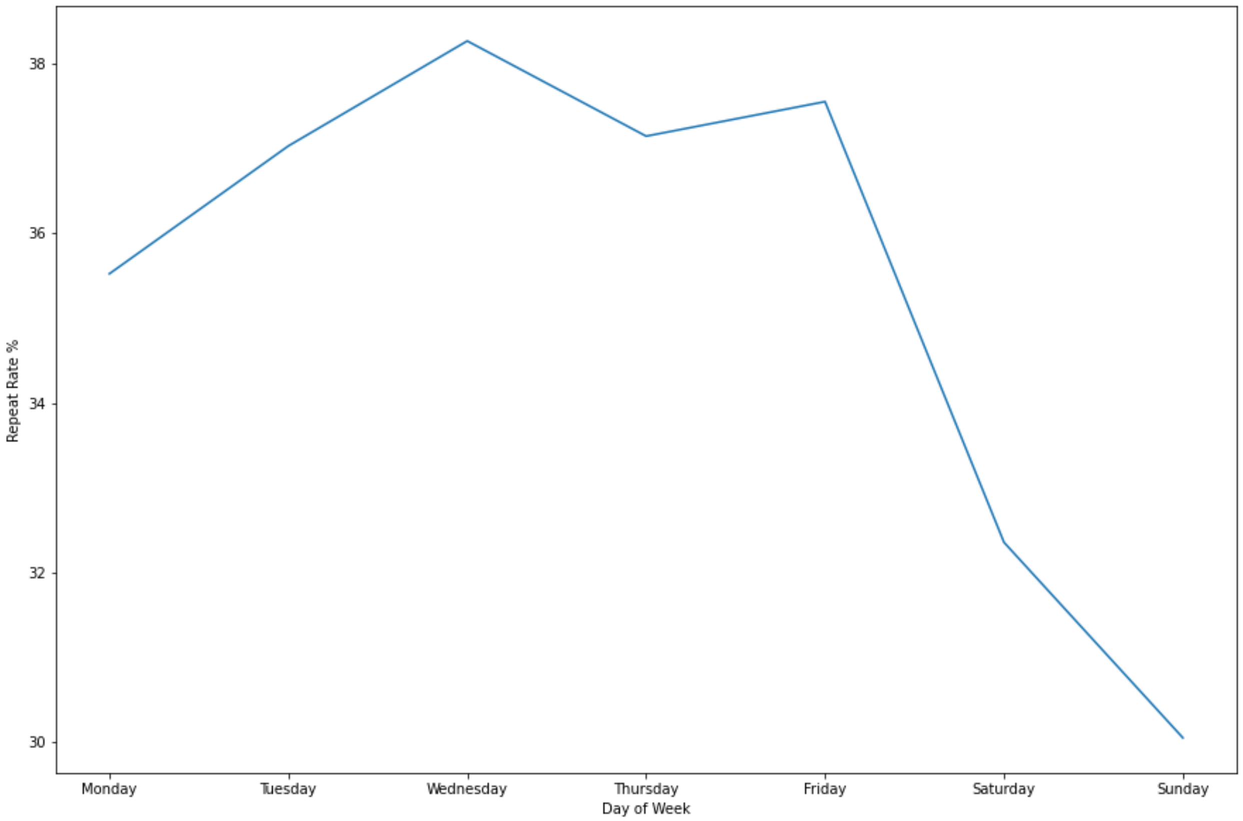


We see that more customers from 2017 and 2018 were retained; 2019 showed a dip in this trend, but based on the data we received (up until September 2020), the following year had not finished yet so that is not a fair rate to compare. This may be something that we segment down more in the future and analyze in more ways, but we stopped analyzing this any further for now.

We also took a quick look at the repeat rate as well. This can often be confused with retention rate, but they are not the same. Repeat rate is the percent (or fraction) of purchases at any particular time segment that are repeat purchases. Retention is more about keeping the customers; repeat is more about when those kept customers are purchasing again. Studying this can hopefully help target when those customers are likely to repeat (relative to overall sales).

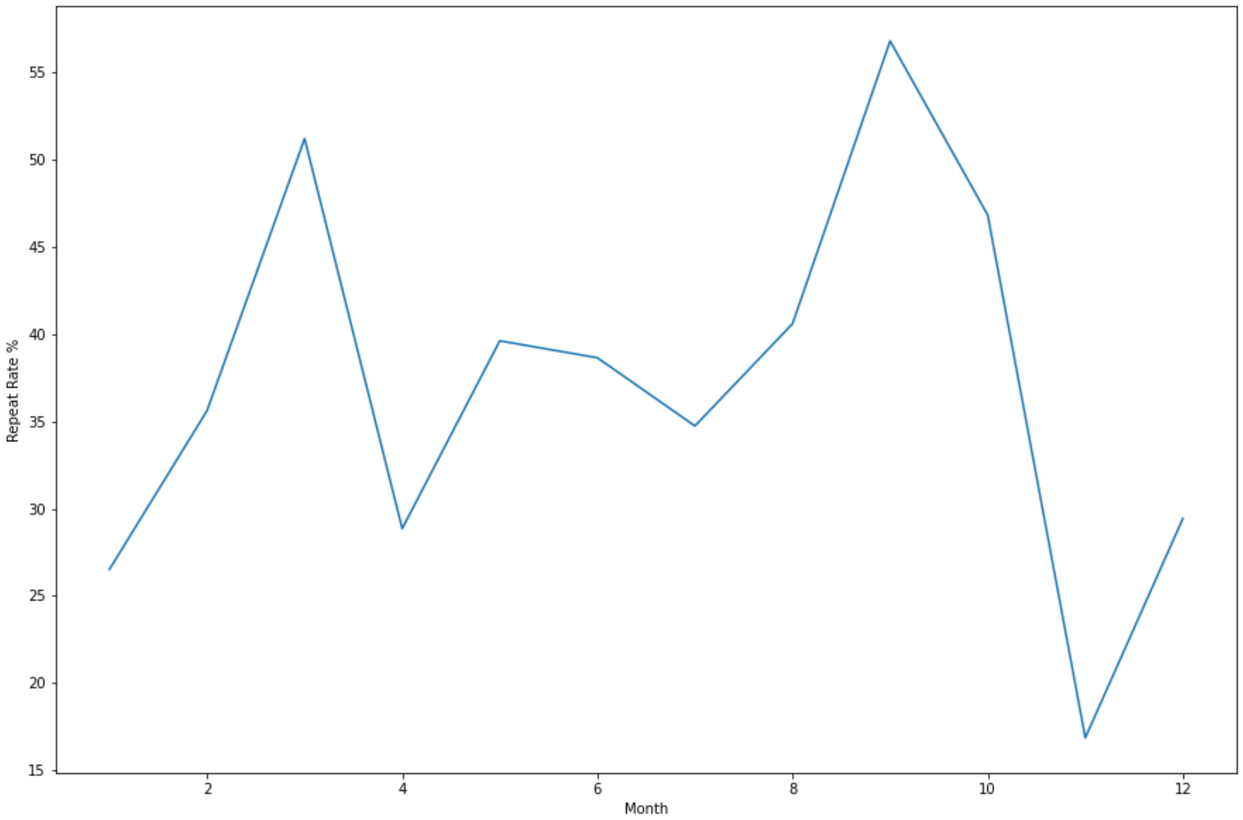


Similar to retention rate, we can see that more of the purchases in 2017 and 2018 were repeat purchases. We see a decline in the amount of repeat purchases in 2019 as more new customers were acquired, but that repeat rate jumps back up in 2020 as a good sign. We also examined the repeat rate as a function of the day of week, as we show in the graph below.



In an earlier graph we saw that customers acquired on Monday and Tuesday tended to spend more over their lifetime. In this plot, we can see that repeat purchases tend to happen more toward the middle and end of the week (Tuesday - Friday). This could be good assistance in determining when to try for new customers (Monday and Tuesday) and when to try to get your existing customers to repeat (Wednesday through Friday).

We'll take one last look at repeat rate before moving on; in this case, we examined repeat rate as a function of monthly sales.



While we acquire more higher value customers in November, it looks like repeat business accounts for very little of overall sales in NOvember. Customers that are repeat purchasers, account for much more of the business in March and September. This could also be examined more closely, but we will move on to modeling for now.

Before moving on to modeling, we extracted all of the repeat information from the first purchase a customer makes and used that as X values to pass on to modeling: 'first_total', 'Marketing_first','first_items', 'first_order', 'server', 'Vendor', 'Emp', 'Source', 'ship_bill', 'Area_Code', 'Ship_Zip', 'lead_sku', 'weekday', 'mon', 'first_tot_lg','first_it_lg', 'boost'. We set our target Y values to the log of the customer lifetime value (‘life_lg’). We move on to pre-processing from here.

Pre-processing

We whittled down our data set to a smaller set of variables for modeling and in this section we prepared that data set for modeling. Specifically, the variables we are working with are:

- first_total: total $ spend on first order
- Marketing_first: whether they accept marketing on the first order
- first_items: number of items on first order
- first_order: date-time of first order
- server: domain name of the customer email

- vendor: 0 = first order from company; 1 = first order from outside source
- Source: web or iphone
- Area_Code: area code of order placed
- Ship_Zip: zip code of shipping address
- lead_sku: name of SKU that was lead item on purchase
- weekday: day of week first order was placed
- mon: month that first order was placed
- first_tot_lg: log of first order total
- first_it_lg: log of number of items in first order

Some of these items are categorical and some redundant. Let's eliminate one of the redundant variables off of the bat: "first _it_log" and "first_items" both are covering the number of items in the first order these are redundant, so we drop "first_it_log" and keep "first_items" (we may try the reverse in the future).

Looking at "Area_Code", there are 373 unique area codes with no area code accounting for more than 0.68% of total orders. While it would be great to aggregate some of these area codes (by region, income, density, or other factors), it's best that we not spend too much time digging into this for now. We dropped this feature. Likewise with zip codes; there were 15,930 unique values and no more than 48 customers in any one of those zip codes. Rather than aggregate, we just dropped that feature as well.

After eliminating those features, that brings us down to the following features:

- server: 26
- source: 6
- lead_sku: 26
- weekday: 7
- mon: 12
- total: 77 more features

That seems reasonable for us to add as features. We created dummy variables for all of these categorical features and split the dataset into train and test datasets (80% / 20% split) before modeling.

Modeling

We whittled down our data set to a smaller set of variables for modeling and in this section, we try a number of different models to predict Customer Lifetime Value (CLV). The first approach to modeling that we take is the most obvious: linear regression. The resulting model had extremely poor performance with Mean Squared Error (MSE) of $4.3x10^{23}$ (Mean Absolute Error of 10,413,803,859.8)and an intercept in the billions. These values seem way off; there may be some very poor features in the model that should be eliminated before running this regression again. We built 2 other types of regression before returning to linear regression with reduced features.

Since linear regression was so off, we chose to try Random Forest Regression (RFR), as it can be more robust to containing too many features. The resulting model has a MSE of 0.068 and MAE of 0.185; that is a much better performing model. In creating this model, we found 25 features that had zero importance in the RFR model; we saved these features to use in feature reduction for the linear regression. Before returning to linear regression, we built a Gradient Boosting Regression model and Dummy model (average).

The Gradient Boosting Regression used all of the features and had a MSE of 0.0681 (MAE of 0.185). We found 39 features in this model that had zero importance; we saved these as well to be removed from the linear regression model.

We based the dummy model on the average of the test sample's y values (log of the CLV); the result was a MAE of 0.291 (Root Mean Squared Error -RMSE of 0.380) and MAE of 0.145. This means that the Random Forest Regressor performed only 0.119 better in terms of RMSE; this is not much when you consider this value close to zero on the log scale, where it only equals $1.32 more in customer lifetime value when considered near zero. However, if we consider that difference close to the mean of CLV, the difference, as the exponential creates a difference of $20.67. We show a table below that summarizes this first pass at modeling.

Model	MAE	MSE	RMSE
Random Forest 1	1.847552e-01	6.807910e-02	2.609197e-01
Gradient Boost 1	1.847552e-01	6.807910e-02	2.609197e-01
Dummy (Average)	2.906511e-01	1.445166e-01	3.801533e-01
Linear Regression 1	1.041380e+10	4.308612e+23	6.564002e+11

The performance of the Linear Regression was horrible on this first round, but we reduced the features and that improved performance drastically.

Reduced Feature Modeling

There were 25 features in the Random Forest Regressor and 39 Features in the Gradient Boosting Regression; we removed all of these features from the data set and performed the train / test split again before modeling. We then built the same models again: Linear Regression, Random Forest Regression, and Gradient Boosting Regression. We even tried an XGBoost. Below is the table that now includes those error values.

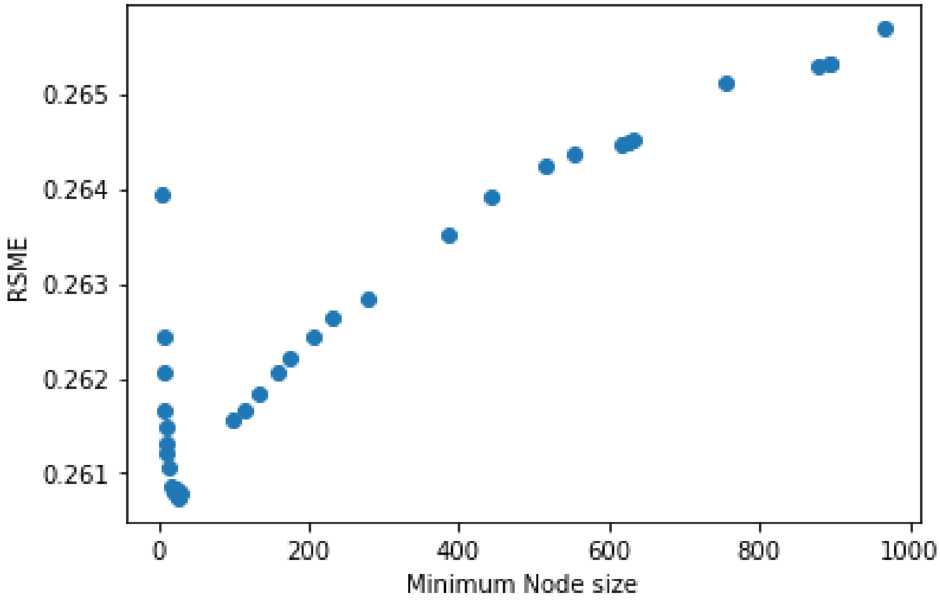
Model	MAE	MSE	RMSE

Linear Regression 2	1.859726e-01	6.770457e-02	2.602010e-01
Random Forest 1	1.847552e-01	6.807910e-02	2.609197e-01
Gradient Boost 1	1.847552e-01	6.807910e-02	2.609197e-01
Random Forest 2	1.847164e-01	6.809799e-02	2.609559e-01
Gradient Boost 2	1.847164e-01	6.809799e-02	2.609559e-01
Dummy (Average)	2.906511e-01	1.445166e-01	3.801533e-01
XG Boost	4.628376e-01	3.046849e-01	5.519827e-01
Linear Regression 1	1.041380e+10	4.308612e+23	6.564002e+11

The Linear Regression model improved significantly after those unimportant features were removed; in fact, it was the top performing model in terms of Mean Squared Error (MSE). The Random Forest and Gradient Boost performed very slightly worse with the reduced features, but that is not until the 4th digit in MSE and MAE.

Hyperparameter Tuning

Since we have ensemble models in Random Forest and Gradient Boost, we tuned those hyperparameters for hopefully better model results. We first looked at tuning just one feature in the Random Forest Regressor: the minimum sames for a leaf. Below is the plot from randomly selecting from a range from 5 to 1000 (with heavier sampling between 5 and 30).



This tuned model performed best with a leaf size of 26, where it had a MSE of 0.06798 and MAE of 0.18597. This is the best overall model when ranked by MAE and 2nd best when ranked by MSE (just behind linear regression).

We also tuned the Gradient Boost by randomly selecting 20 values for the learning rate between 0 and 0.4. That hyperparameter alone did not have any effect on the model performance - it performed exactly the same regardless of learning rate (and exactly the same as the Gradient Boost 2 (reduced features) listed in the above table.

More Extensive Hyperparameter Tuning

The hyperparameter tuning that we detail above was just on one parameter for each model type. We also performed a more extensive random search over multiple hyperparameters using RandomizedSearchCV to also cross validate. For the Random Forest Regressor, we used a variety of values for: the number of estimators (200 to 2000); max depth (10 to 110); minimum samples for a split (5 to 40); minimum leaf size (5 to 40). The best performing model in that randomized search had 600 estimators, 26 samples at a split, 5 minimum leaf samples, and a maximum depth of 20. However, when this model was tested on the hold-out data, it didn't perform as well as the Random Forest Regressor model above that was just tuned on one hyperparameter (leaf size).

Model	MAE	MSE	RMSE
Linear Regression 2	1.859726e-01	6.770457e-02	2.602010e-01
RF Tune 1	1.839021e-01	6.798485e-02	2.607391e-01
Random Forest 1	1.847552e-01	6.807910e-02	2.609197e-01
Gradient Boost 1	1.847552e-01	6.807910e-02	2.609197e-01
Random Forest 2	1.847164e-01	6.809799e-02	2.609559e-01
Gradient Boost 2	1.847164e-01	6.809799e-02	2.609559e-01
RF Hypertune	1.847757e-01	6.767430e-02	2.609559e-01
GB Hypertune	1.832013e-01	6.889463e-02	2.624779e-01
Dummy (Average)	2.906511e-01	1.445166e-01	3.801533e-01
XG Boost	4.628376e-01	3.046849e-01	5.519827e-01
Linear Regression 1	1.041380e+10	4.308612e+23	6.564002e+11

We also performed similar hyperparameter tuning on the Gradient Boost model, tuning: number of estimators (200 to 2000); learning rate (0.0001 to 0.4); minimum samples for a split (2 to 40); minimum sample for a leaf (1 to 40). The best model had 600 estimators, 26 sample minimum at splits, 1 sample minimum leaf size, a maximum depth of 10, and a learning rate of 0.02. We highlighted these results in the previous table - this did not perform better on the hold-out test data than Gradient Boosting with just the default parameters. It turns out the default parameters for both Random Forest and Gradient Boosting were pretty good to begin with.

Ensemble Model: Linear Regression + Random Forest Regression

To improve model performance further, we tried stacking two models: using linear regression first and then using Random Forest Regression on the residuals from that linear regression. These were the two best models in terms of performance (linear regression 2 and RF Tune 1), so we hoped that we could produce an even better model by combining them. We added the result of these models to the table below.

Model	MAE	MSE	RMSE
Ensemble: LR2 + RF	1.821710e-01	6.680951e-02	2.584753e-01
Linear Regression 2	1.859726e-01	6.770457e-02	2.602010e-01
RF Tune 1	1.839021e-01	6.798485e-02	2.607391e-01
Random Forest 1	1.847552e-01	6.807910e-02	2.609197e-01
Gradient Boost 1	1.847552e-01	6.807910e-02	2.609197e-01
Random Forest 2	1.847164e-01	6.809799e-02	2.609559e-01
Gradient Boost 2	1.847164e-01	6.809799e-02	2.609559e-01
RF Hypertune	1.847757e-01	6.767430e-02	2.609559e-01
GB Hypertune	1.832013e-01	6.889463e-02	2.624779e-01
Dummy (Average)	2.906511e-01	1.445166e-01	3.801533e-01
XG Boost	4.628376e-01	3.046849e-01	5.519827e-01
Linear Regression 1	1.041380e+10	4.308612e+23	6.564002e+11

The ensemble model resulted in the best performance out of all of the models that we tried, having both the least MAE and MSE.

Model Validation

The best models in the table may just be a product of the particular test / train data split, so we went back and tried a different test / train split to validate the top models.

Model	MAE	MSE	RMSE
Ensemble: LR2 + RF	0.182427	0.068368	0.261472
Random Forest Tune	0.184347	0.069469	0.263569
Gradient Boost	0.184716	0.068098	0.260956
Linear Regression	0.185567	0.069419	0.263475
XG Boost	0.467911	0.311898	0.558478

These results uphold our previous conclusions that the Linear Regression plus Random Forest (minimum leaf size = 26) is the best model. The other models with just Random Forest and Linear Regression performed well in this test as well with Gradient Boost performing better than in the previous test / train split. We would recommend the Ensemble Model of Linear Regression plus Random Forest, but for simplicity sake, the Random Forest tuned and Linear Regression alone provide good models that performed well on both test and train splits.

Recommendations

The best performing model for estimating Customer Lifetime Value (CLV) was an ensemble model with Linear Regression applied first and then Random Forest Regressor (with minimum leaf size of 26) applied to the residuals. This produced predictions with a Mean Absolute Error (MAE) of 0.1822 in the first train and test set and MAE of 0.1824 in the second split we tried. This seems like a model that performs consistently. If we compare that to the dummy model's RMSE, there is a 0.122 difference. Since this is on an exponential scale, that error can change drastically as the values get larger. Near the mean, it's almost a $20 difference and toward the high end of the model range, it's a $1000 difference (10^(3.5 + 0.122)). The model is particularly useful when it comes to the higher end clients.

Also on a practical note, we pointed out some useful insights in our exploratory data analysis with hypothesis testing to back up those close:

- Customers acquired on Monday spend significantly more over their lifetime than customers acquired on Sunday
- Customers acquired in November spend significantly more over their lifetime than customers acquired in September
- There is significantly more repeat orders in September than in November
- There are significantly more repeat order on Wednesday than on Saturday or Sunday

All of these numbers may be affected by details and timing of the marketing campaigns. It would be great to A/B test these to see if these nuances could be used to generate more sales.

We pulled in over 120,000 rows of data from 5 CSV files into a single dataframe; we then anonymized and cleaned the data. We explored some important trends in customer purchasing for repeat, retention, and overall Customer Lifetime Value. We modeled the Customer Lifetime Value using over a dozen different models to produce results that were better (MAE of 0.182 and RMSE of 0.258) than the dummy model of just assuming the average.

Future Improvements

While we were exploring this data and building the models, we made some data and modeling decisions for our particular models. However, we did notice that we could take different approaches that may improve results:

1. **Any SKU from the first order** - We built our models based on the lead SKU in the first order, but it's possible that other significant items may be buried further down in the first order that are very telling about the future purchases of that customer. If we were to expand this model, we would consider all of the SKU's in a customer's first order.
2. **Model the future CLV** - Our models were built on the total customer lifetime value, but some of that value has already occurred in the first purchase. The more important question for a company is "how much are they going to spend in the future?" If I were to build the models again, I would subtract the first order from the Customer Lifetime Value to get a Customer Future Value. With that information and a first order placed, the company could evaluate how much business they can expect from a customer in the future.
3. **Aggregates of zip or area code** - I dropped zip and area codes from the model because there were too many of them without any particular one sticking out. If I were to approach this with more time, it would be possible to aggregate these categorical variables based on many potential values and see if they had any impact on Customer Lifetime Value. Some initial ways of aggregate the data would be on geographic location, income, population, and density.
4. **Bins for time of day** - We used the days of the week and month of the year as features, but we never investigated to see if the time of day that somebody purchased had a large impact on CLV. That would be interesting to create bins for the different times of day to see if that had an impact on their future purchases.
5. **Lasso and Ridge Regression** - There are still a lot of features that were included in the model; we might want to take their regularization approaches to reduce those features and get a cleaner model with lower coefficient values. We may want to even include this in an ensemble approach.

6. **Neural Net** - I started composing a stacked model, with linear regression and then random forest regression, but I did not build a full neural network. It could be possible to build a neural network that improves with every iteration and additional information.
7. **Box - Cox transformation on data instead of log10** - I transformed the data using log10, but I could have also tried a Box-Cox transformation to see if that method produced less skew in the data.
8. **Log10 of more features** - I took the log of the first purchase value but not of some other features that were exponentially distributed: number of items in the first order comes to mind.

All of these could provide improvements to the model. With this much data to investigate, there is a lot more exploration I could do with this data as well. Thank you for reviewing this analysis and modeling of Shopify Sales Data. I hope this information proves useful.