Customer Lifetime Value Prediction

Springboard Data Science Career Track

Introduction

Customer Lifetime Value (CLV) is the total worth to a business of a customer over the whole period of their relationship. It is one of the most important marketing metrics as it provides crucial guidance for customer acquisition initiatives and even overall marketing strategies.

CLV is commonly calculated based on historical data. Historical CLV will provide specific insights on what has happened before, but it fails to shed a light on the future lifetime value of the customers.

Predictive CLV incorporates the expected future behavior of the customers and provides a better tool for businesses to make their marketing decisions.

Problem Statement

We will build a supervised classification machine learning model and a regression machine learning model to predict existing customers' lifetime value based on their online shopping behavior. For the classification model, we will segment customers into three groups: low, mid and high lifetime value, which provides more actionable insights for businesses.

Online retailers can use our customer LTV prediction model to extract the true values of their customers, and allocate their marketing budget and efforts intelligently.

Data Source

The dataset contains all the transactions occurring between 01/12/2009 and 09/12/2011 for a UK-based and registered non-store online retail. The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers. The dataset was retrieved from the UCI Machine Learning Repository, and contains 1,067,371 rows and 8 columns.

The variables include:

- Invoice - ID unique to each transaction. If this code starts with the letter 'c', it indicates a cancellation.
- StockCode - ID unique to each product.
- Description - Product name.

- Quantity - The quantities of each product per transaction.
- InvoiceDate - Invoice date and time.
- UnitPrice - Product price per unit in sterling (£).
- CustomerID - ID unique to each customer
- Country - The name of the country where a customer resides.

Data Wrangling

Duplicates and Missing Values

After removing duplicate rows, the dataset has over 235k rows missing the customer ID, ideally we should try to fill out the missing values. However, since we do not have the information required to fill in the missing Customer ID, we will remove these rows and now we are left with 797k rows.

Test and Postage Transactions

There are some test order and postage transactions in the dataset, and they should not be included in the customer lifetime value prediction. We will remove them too.

Item Total Attribute

We will add a column called Item Total to the dataset, which is the total revenue for an item in that specific transaction.

Outliers

By examining the box plots of quantity, price, and item total columns (Figure 1), we see that the ranges of these attributes are quite wide, with the highest price being almost £40,000, which is highly unlikely considering the nature of the business.

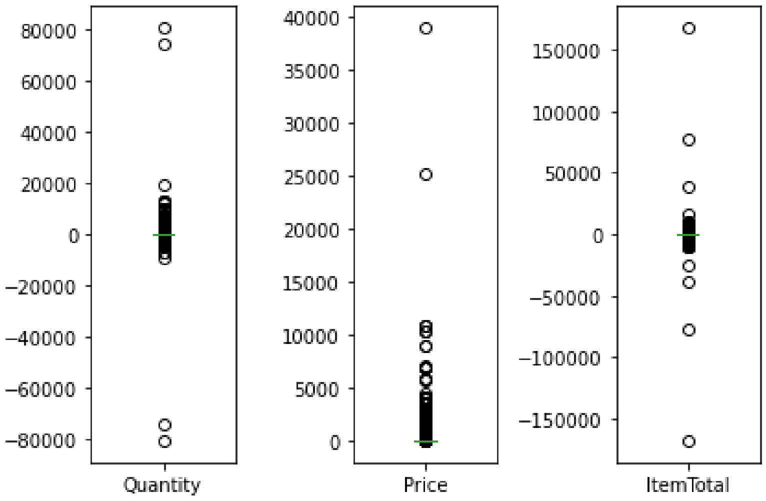


Figure 1. Box plots of Quantity, Price, and Item Total

We can see a lot of transactions with high prices are manual inputs for cancellations. Now that we have a general idea about what these "outliers" are, and we know they are meaningful for the customer LTV prediction, we will keep them for now.

Exploratory Data Analysis

Distribution of Order Total

We now want to further explore the data and look at the distribution of order total for all orders. Figure 2 shows the distribution of order totals between -£40,000 and £40,000 to exclude outliers.

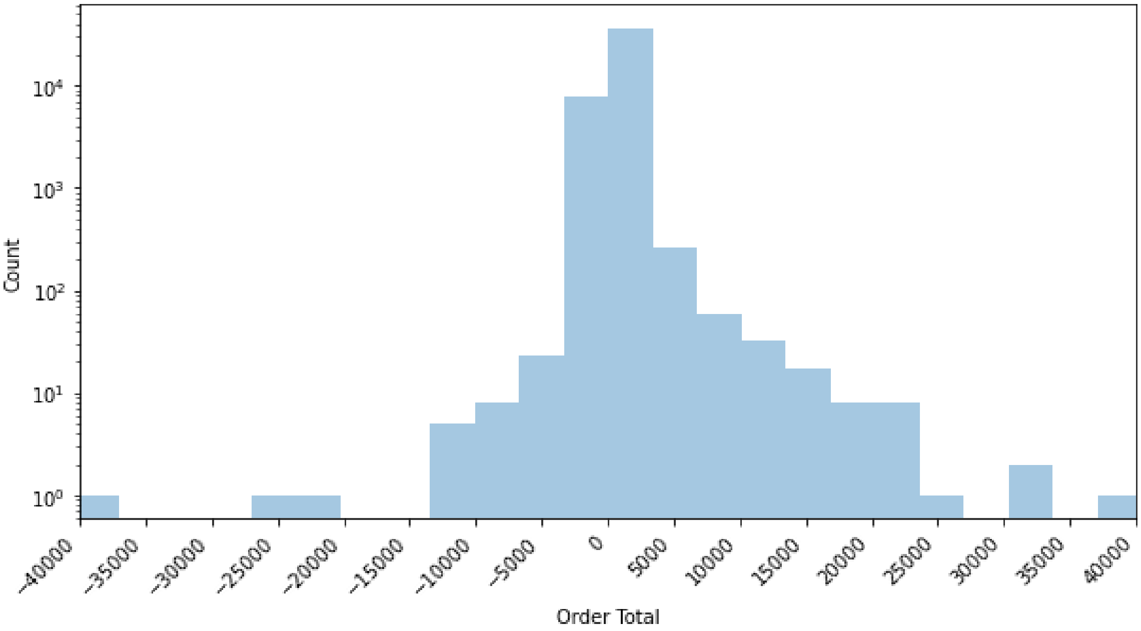


Figure 2. Histogram of Order Total

Order Total for each Country

We are also interested in if any trend or pattern can be observed for order totals when we group them by countries. Figure 3 shows the box plots for all countries in the dataset. The plots are sorted by the average order value for each country from high (left) to low.

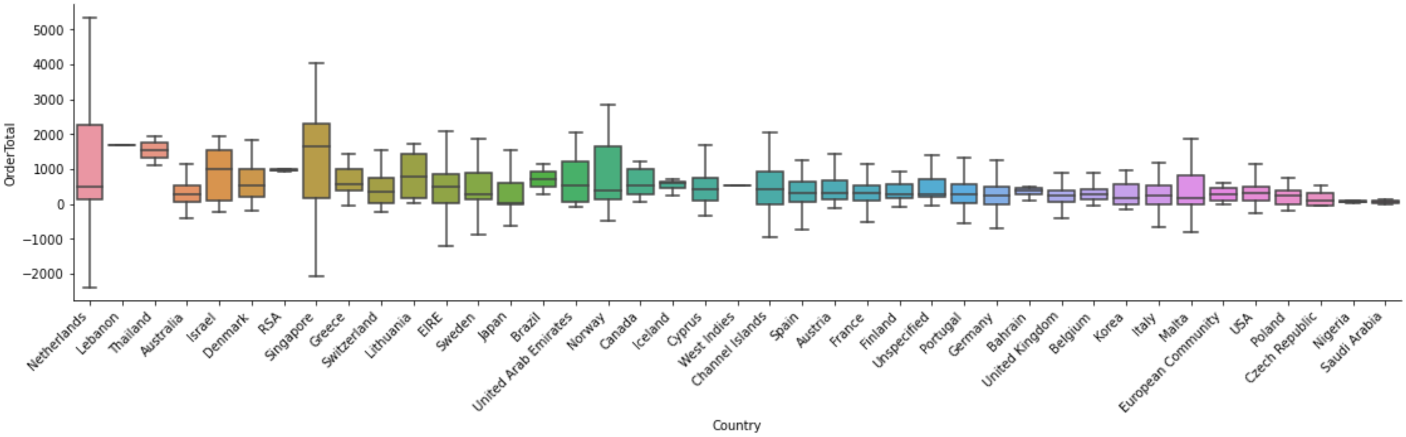


Figure 3. Box Plots for Order Totals Grouped by Countries

Hypothesis Testing

We know that UK and Germany are the top 2 countries where most of the orders are coming from. From the dataset, the average order value of Germany is $20 higher than that of UK, and we are interested in the true difference in the average order values of these two countries.

We form below hypothesis:

- H_0: UK and Germany have the same average order value
- H_1: UK and Germany have different average order values

We use two methods for the hypothesis testing. Neither of their p-values is statistically significant to reject the null hypothesis that UK and Germany have the same average order values, using 0.05 as the α:

- Bootstrapping p-value: 0.28
- T-test p-value: 0.29

Classification Model

Feature Engineering

The dataset has a total of 24 months of sales data. We take 3 months of data to create features, and use them for predicting the next 21 months LTV.

RFM score is a very common and useful approach when predicting customer lifetime value:

- R - Recency
- F - Frequency

- M - Monetary Value

We calculate the Recency, Frequency and Monetary Value for each customer, and apply k-means clustering to identify different groups (scores) for each customer, then sum up the frequency, recency, and monetary scores to get a RFM score.

We create dummy features for the country column to include demographic information into our features.

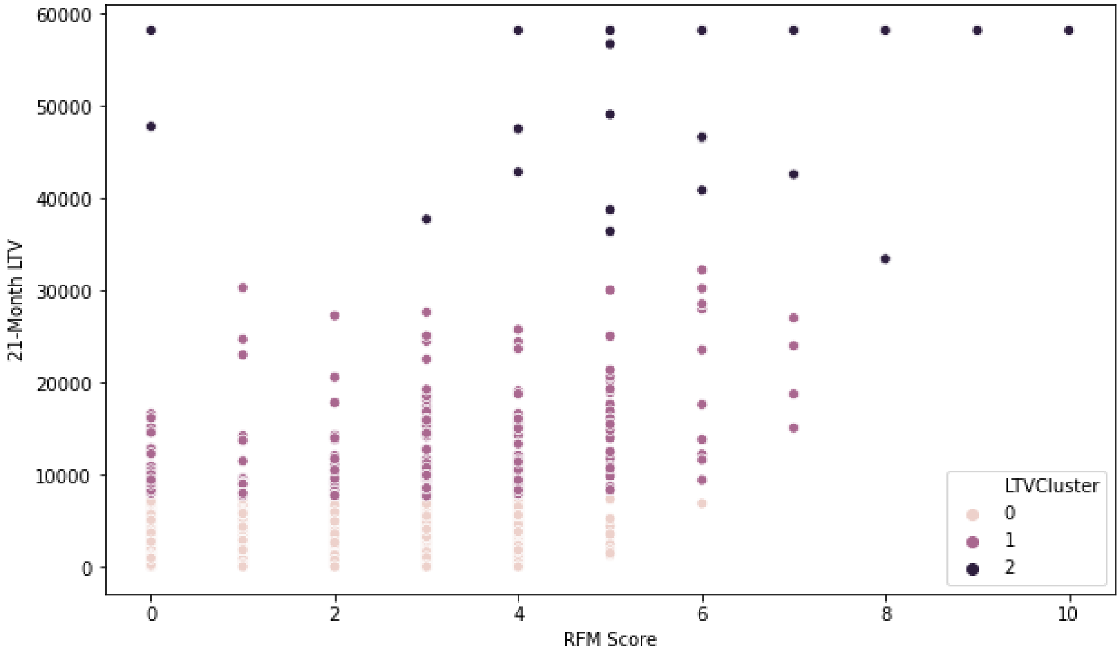
The Label - LTV Cluster

LTV is a regression problem, but here we create LTV segments, and predict which LTV segment the customer belongs to. Having LTV segments will allow businesses to extract more actionable insights and treat customers differently based on their predicted LTV segments.

For this project we apply k-means clustering to identify 3 segments:

- Cluster 0: Low LTV
- Cluster 1: Mid LTV
- Cluster 2: High LTV

We can observe some correlation between the RFM score and the LTV segment from the scatter plot, even though not prominent. Majority of customers with lower RFM scores tend to belong to the low and mid LTV clusters.



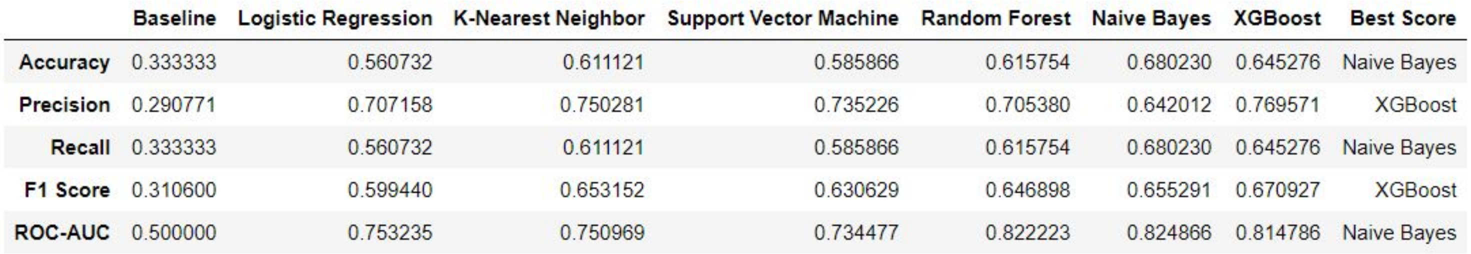
Feature 4. Scatter Plot of RFM Score vs 21-Month LTV

Model Training and Selection

This is a classification problem in supervised learning. In our research we evaluate below classification algorithms:

- Logistic Regression
- K-Nearest Neighbors (KNN)
- Support vector machine (SVM)
- Random Forest
- Gaussian Naive Bayes
- Gradient Boost

To make sure that all classes contribute equally to the average, we choose macro average for the metrics. From the model performance comparison table (Table 1), we can see that Gaussian Naive Bayes and XGBoost models are more effective in predicting the LTV clusters.



	Baseline	Logistic Regression	K-Nearest Neighbor	Support Vector Machine	Random Forest	Naive Bayes	XGBoost	Best Score
Accuracy	0.333333	0.560732	0.611121	0.585866	0.615754	0.680230	0.645276	Naive Bayes
Precision	0.290771	0.707158	0.750281	0.735226	0.705380	0.642012	0.769571	XGBoost
Recall	0.333333	0.560732	0.611121	0.585866	0.615754	0.680230	0.645276	Naive Bayes
F1 Score	0.310600	0.599440	0.653152	0.630629	0.646898	0.655291	0.670927	XGBoost
ROC-AUC	0.500000	0.753235	0.750969	0.734477	0.822223	0.824866	0.814786	Naive Bayes

Table 1. Model Performance Comparison Table

It's important for businesses to identify their high LTV customers and establish specific marketing strategies for these customers. To further compare the Gaussian Naive Bayes and XGBoost models, we plot the confusion matrix for both (Figure 5).

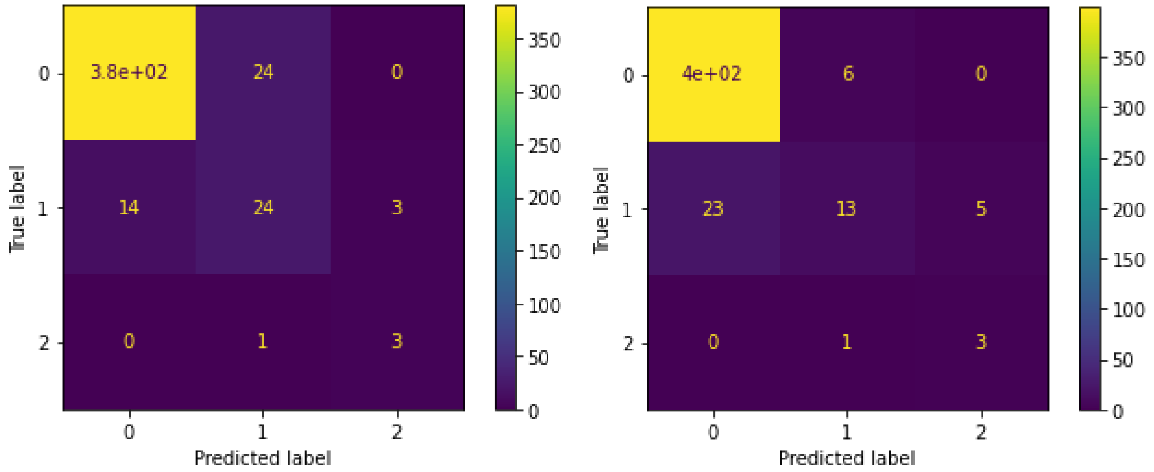


Figure 5. (Left) Confusion Matrix for the Gaussian Naive Bayes Model
(Right) Confusion Matrix for the XGBoost Model

We can see that both models successfully identified 75% of actual high LTV (cluster 2) customers (recall), but Naive Bayes did a better job identifying mid LTV (cluster 1) customers. We will choose the Gaussian Naive Bayes model as the final model.

Future Improvements

The final model only correctly identified 59% mid LTV customers in our test dataset. There is certainly room for improvement:

- Add more features and improve feature engineering
- Add more data to the model if possible

There are also various techniques we can adopt to improve the model performance of imbalanced datasets, for example:

- Under-sampling: Remove samples from over-represented classes
- Over-sampling: Add more samples from under-represented classes

Regression Model

Feature Engineering

After developing a classification model for customer LTV prediction, we are interested in approaching the same question in a regression setting. We will use *lifetimes* - a Python package that has APIs for the models that we're about to use, plus some useful utility functions.

We are still going to use RFM as the features, but this time using a built-in function from *lifetimes* package to transform the dataset into the features we need.

Model Training - BG/NBD Model

We first build a BG/NBD Model to predict the number of repeat purchases up to time for each customer

The basic idea of the BG/NBD model is that sales of each customer can be described as a combination of his/her probability to buy and to churn. it models the sales for a particular customer as a function of 2 distributions - Gamma for transactions and probability of churn as Beta.

From the scatter plot output (Figure 6), we can see that the model's predictions are decent.

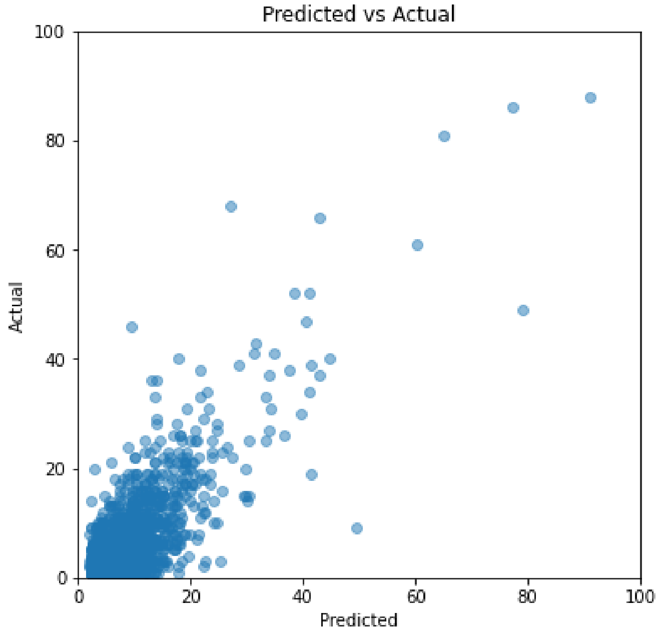


Figure 6. Scatter Plot of Predicted Repeat Purchases vs Actual Repeat Purchases

Figure 7 groups all customers in the calibration period by their number of repeat purchases (x-axis) and then averages over their repeat purchases in the holdout period (y-axis). As we can see, despite our model not fitting the actual purchases perfectly, it was able to capture trends and significant turning points.

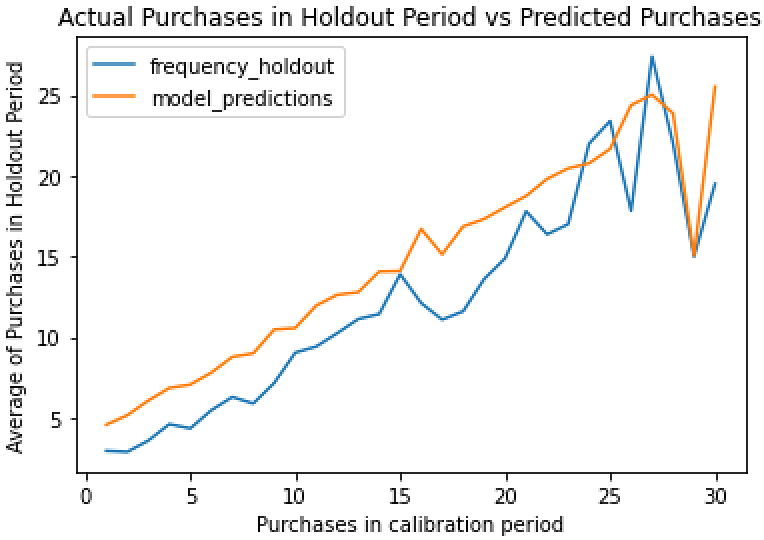


Figure 7. Actual Purchases in Holdout Period vs Predicted Purchases

Gamma-Gamma Model

For the second step, we will use the Gamma-Gamma submodel to predict the spend per transaction in the future at the customer level. This model relies on an important assumption that there is no relationship between the monetary value and the purchase frequency.

We then use a built in function to predict customer LTV. This function will take the prediction of expected purchase and combine it with the expected purchase value. These components allow us to arrive at an estimate of how much a customer is worth in a given period of time. Figure 8 gives us a general idea of how well does the model perform.

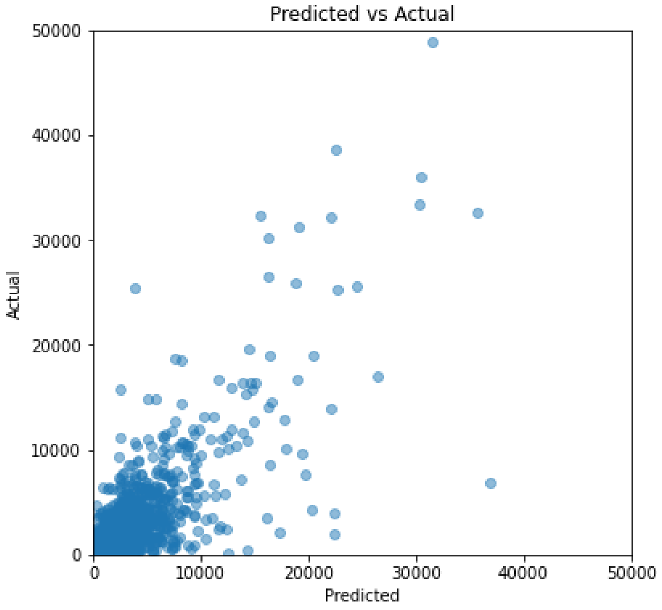
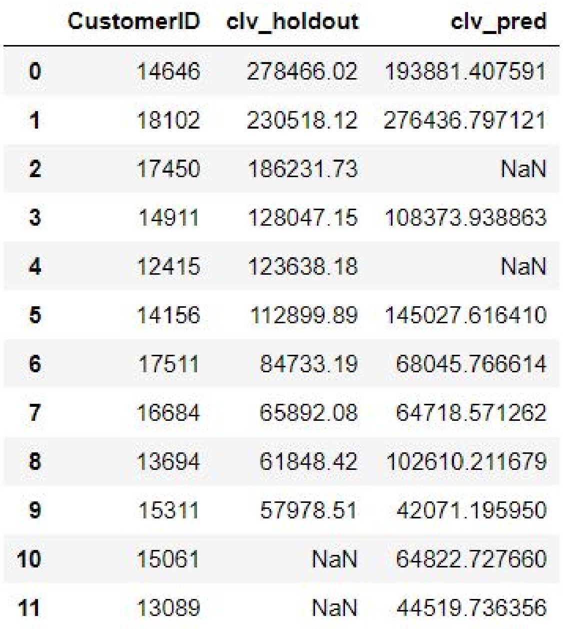


Figure 8. Scatter Plot of Predicted LTV vs Actual LTV

Now we can see the predicted top 10 most valued customers, and compare the prediction with the actual top 10 most valued customers (Table 2).



	CustomerID	clv_holdout	clv_pred
0	14646	278466.02	193881.407591
1	18102	230518.12	276436.797121
2	17450	186231.73	NaN
3	14911	128047.15	108373.938863
4	12415	123638.18	NaN
5	14156	112899.89	145027.616410
6	17511	84733.19	68045.766614
7	16684	65892.08	64718.571262
8	13694	61848.42	102610.211679
9	15311	57978.51	42071.195950
10	15061	NaN	64822.727660
11	13089	NaN	44519.736356

Table 2. Top 10 Actual vs Predicted Most Valued Customer

To conclude, we can see that the model was able to capture the majority of the top 10 most valued customers, and only mispredicted 2 customers.

This model jointly models the probability to churn, purchase, and the average purchase value. It takes a few simple features, and is quite accurate when we look at the aggregated level.

Future Improvements

Due to its simplicity, this model probably underfits the data and has inferior predictive performance compared to the more complex machine learning models. Additionally, it cannot take context data (e.g. customer demographics) into account, which also limits the predictive power of the model.

For the next steps of this project, these are some of the improvements we can make:

- Fit the model on customer cohorts (ex. split by user country).
- Join the model with a linear model with additional features (ex. a customer's website visits, time since last visit, product reviews, channel of acquisition, etc.).