Measuring Customers Value Using Python/Lifetimes: an example with complete Python code

Sarit Maitra Sep 25, 2019 · 9 min read ★

CLTV analysis and customer-centric approach to marketing has lead to a success story in quantitative analytics with the increasing availability of transaction data. A number of studies have developed scoring mechanisms (e.g. regression models to predict a customer's future behavior). The measures of a customer's past behavior are key predictors of their future behavior in all the empirical analysis. It is common practice to summarize a customer's past behavior by investigating the "RFM" characteristics: recency (time of most recent purchase), frequency (number of past purchases), and monetary value (average purchase amount per transaction). However, empirical studies reveal the limitations with these models when seeking to develop CLTV estimates.

> Not all customers are equally important to a firm. Maintaining long-term relation with all of them (especially the loss makers) is not optimal because eventually marketing is all about attracting and retaining profitable customers (Kotler and Armstrong 1996). Hence the objective of CLTV is firstly on general topics of firm's profitability and secondly as an input in customer acquisition decision and customer acquisition/retention trade-offs (Berger and Nasr 1998).

Objectives

The objectives here is to build a probabilistic model for predicting CLTV in non-contractual setting on an individual level. Using the results of this exercise, managers should be able to (a) Distinguish active customers from inactive customers, (b) Generate transaction forecasts for individual customers, (b) Predict the purchase volume of the entire customer base.

Approach

Lifetimes links the RFM (recency, frequency, monetary value) paradigm with customer lifetime value (CLTV). The stochastic model presented here, featuring BG/NBD

framework to capture the flow of transactions over time. BG/NBD portrays the story being about how/when customers become inactive.

The BG/NBD require only two pieces of information about each customer's past purchasing history: "recency" (when the last transaction occurred) and "frequency" (how many transactions was made in a specified time period). The notation used to represent this information is [X = x, t(x), T], where x is the number of transactions observed in the time period (0, T) and t(x) ($0 < t(x) \leq T$) is the time of the last transaction. Using these two key summary statistics, SMC(2) derive expressions for a number of managerially relevant quantities, such as:

> *E[X(t)], the expected number of transactions in a time period of length t, which is central to computing the expected transaction volume for the whole customer base over time.*
>
> *P[X(t) = x], the probability of observing x transactions in a time period of length t.*
>
> *E[Y (t) | X = x, t(x), T], the expected number of transactions in the period (T,T + t] for an individual with observed behavior (X = x, tx, T).*

Therefore, customers will purchase at a randomly distributed interval within a time range. After each purchase they have a certain probability of dying or becoming inactive. Each customer is different and have varying purchase intervals and probability of going inactive.

Let's explore the data.

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

BG/NBD Model (CLTV analysis)

The following nomenclature is used: Frequency (F) is the number of repeat purchases the customer has made. T represents the age of the customer which is equal to the duration between a customer's first purchase and the end of the period under study. Recency (R) is the age of the customer when they made their most recent purchases.

This is equal to the duration between a customer's first purchase and their latest purchase. There is one last component called Monetary Value that doesn't quite come in until later. These four components together is called an RFM Matrix.

After doing the necessary cleaning and creating a new data frame containing CustomerID, InvoiceDate (remove the time) and adding a new column ('sales') :

```
import datetime as dt
data['InvoiceDate'] = pd.to_datetime(data['InvoiceDate']).dt.date
data = data[pd.notnull(data['CustomerID'])]
data = data[(data['Quantity']>0)]
data['Sales'] = data['Quantity'] * data['UnitPrice']
cols_of_interest = ['CustomerID', 'InvoiceDate', 'Sales']
data = data[cols_of_interest]
print(data.head())
print(data['CustomerID'].nunique())
```

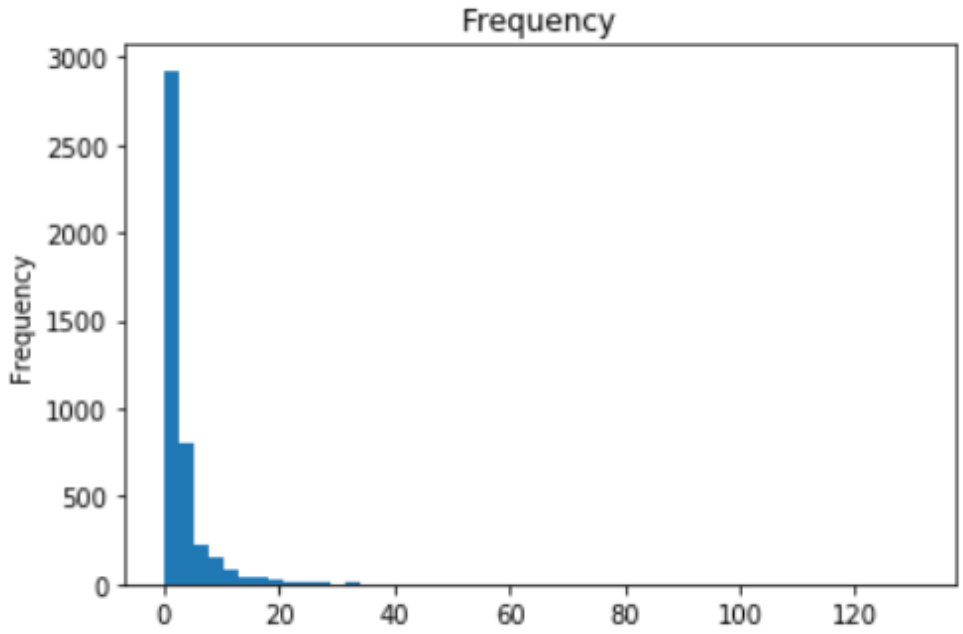
```
   CustomerID InvoiceDate  Sales
0     17850.0  2010-12-01  15.30
1     17850.0  2010-12-01  20.34
2     17850.0  2010-12-01  22.00
3     17850.0  2010-12-01  20.34
4     17850.0  2010-12-01  20.34
4339
```

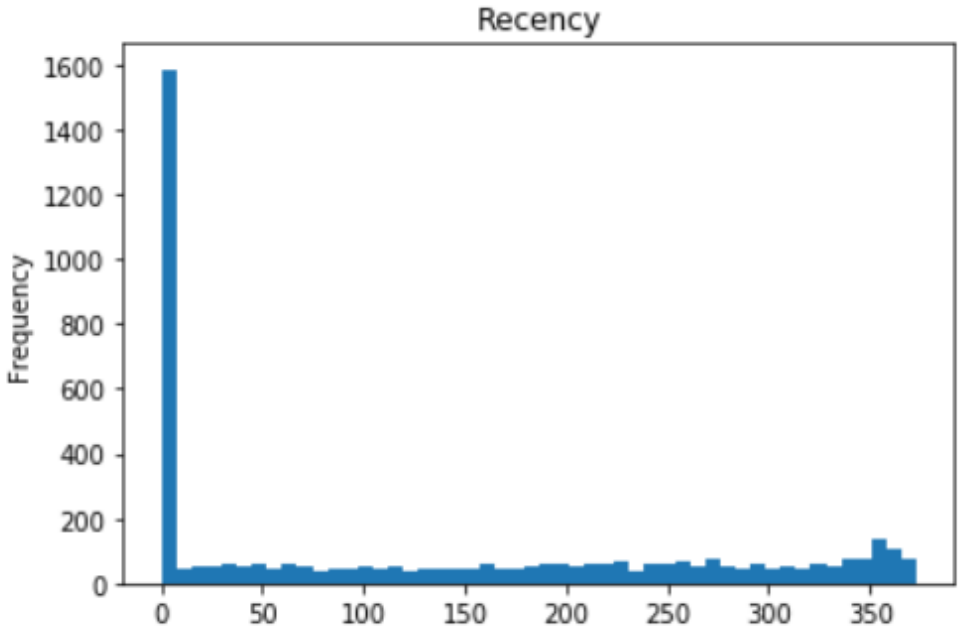
```
df = summary_data_from_transaction_data(data, 'CustomerID', 'InvoiceDate', monetary_value_col='Sales', observation_period_end='2011-12-9')
df.head()
```

CustomerID	frequency	recency	T	monetary_value
12346.0	0.0	0.0	325.0	0.000000
12347.0	6.0	365.0	367.0	599.701667
12348.0	3.0	283.0	358.0	301.480000
12349.0	0.0	0.0	18.0	0.000000
12350.0	0.0	0.0	310.0	0.000000

We can make some observations here. There are 4339 customers and 12346 made single purchase, so the F and R are 0, and the T is 325 days.

```
df['frequency'].plot(kind='hist', bins=50)
print(df['frequency'].describe())
print(sum(df['frequency'] == 0)/float(len(data)))
```





	Frequency	Recency
count	4339.000000	4339.000000
mean	2.864024	130.741415
std	5.952745	132.210176
min	0.000000	0.000000
25%	0.000000	0.000000
50%	1.000000	93.000000
75%	3.000000	252.000000
max	131.00000	373.0000

As shown, both frequency and recency are distributed quite near 0. Among all customers, >38% of them only made zero repeat purchase while the rest of the sample (62%) is divided into two equal parts: 31% of the customer base makes one repeat purchase while the other 31% of the customer base makes more than one repeat purchase. Similarly, for Recency, most customers have made their last purchase early in their lifetime and then became inactive. Indeed, the last repeat purchase that half our customers will make is within less than a year (252 days to be precise), since their first purchase for 75th quantile.

We will develop the beta-geometric/NBD (BG/NBD). BG/NBD represents a slight variation in the behavioral story associated with the Pareto/NBD, but it is quite easy to implement. If any one interested to read more about BG/NBD, may read this article.

Let's do the Model talking now

We first need to fit the customer probability model to the data so that it picks up on their behaviors and pattern. This is done by looking at each individual's Frequency, Recency and Age and adjusting its parameters so that it better reflects the intervals in which our customer-base purchases.

Mathematical Box: Likelihood Derivation

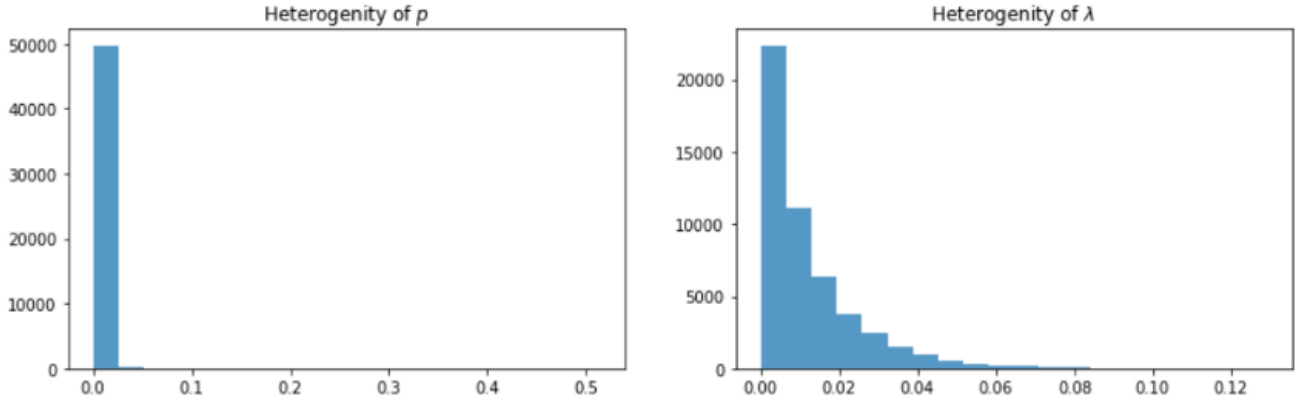
The parameters also vary across different customers so it is calculated over two distributions for a more accurate and flexible fit of the data. Mathematically, this is done by taking the expectation of the equation over both distributions.

```
from scipy.stats import gamma, beta
bgf = BetaGeoFitter(penalizer_coef=0.0)
bgf.fit(df['frequency'], df['recency'], df['T'], )
print (bgf)

# Plot
gbd = beta.rvs(bgf.params_['a'], bgf.params_['b'], size = 50000)
```

```
ggd = gamma.rvs(bgf.params_['r'], scale=1./bgf.params_['alpha'], size
= 50000)
plt.figure(figsize=(14,4))
plt.subplot(121)
plt.title('Heterogenity of $p$')
temp = plt.hist(gbd, 20, alpha=0.75)
plt.subplot(122)
plt.title('Heterogenity of $\lambda$')
temp = plt.hist(ggd, 20, alpha=0.75)
```

```
<lifetimes.BetaGeoFitter: fitted with 4339 subjects, a: 0.00, alpha: 68.89, b: 6.75, r: 0.83>
```



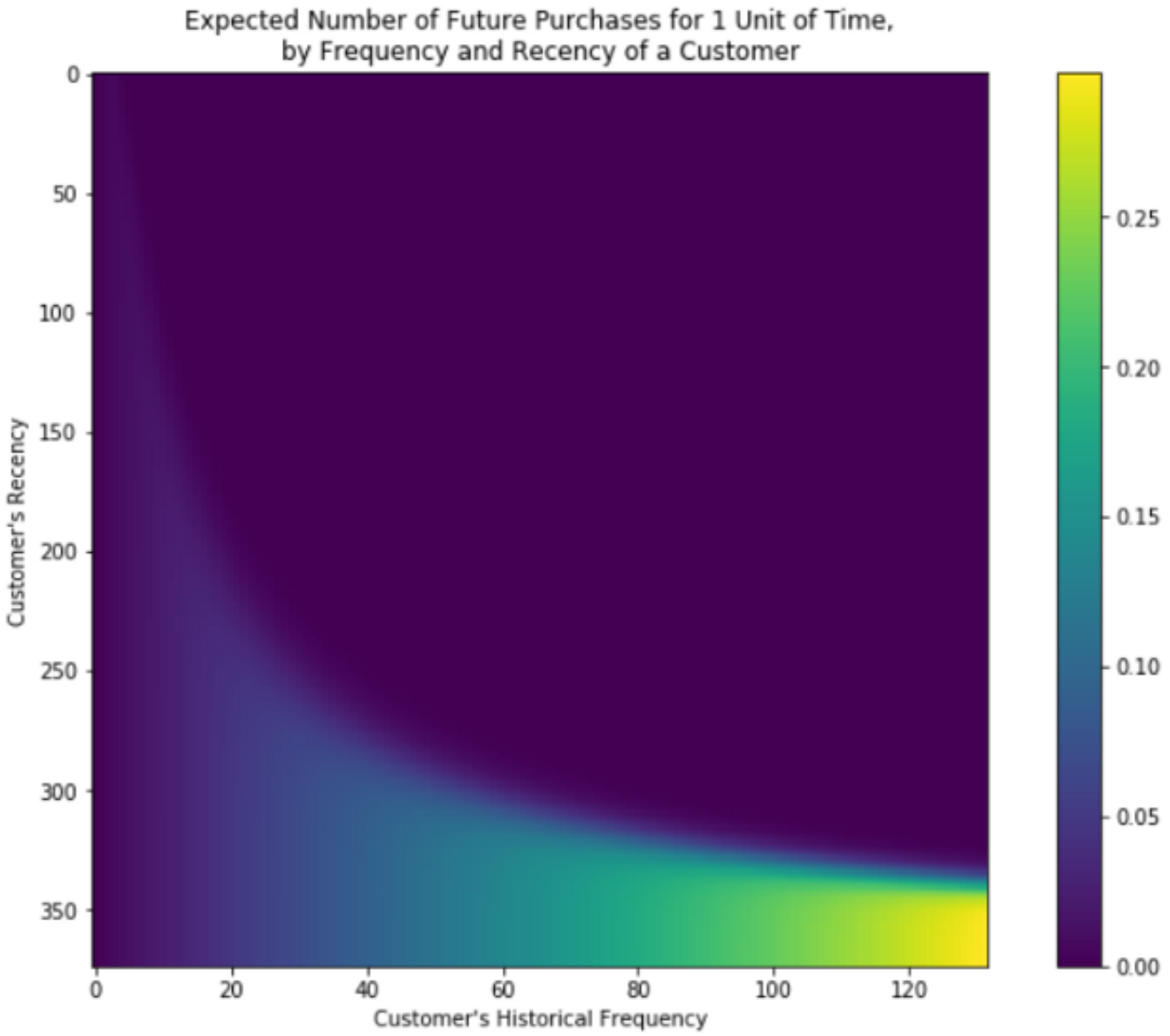

```
1 bgf.summary
```

	coef	se(coef)	lower 95% bound	upper 95% bound
r	0.826433	0.026780	0.773944	0.878922
alpha	68.890678	2.611055	63.773011	74.008345
a	0.003443	0.010347	-0.016837	0.023722
b	6.749363	22.412933	-37.179985	50.678711

Visualizing F/R Matrix

Frequency/Recency matrix computes the expected number of transactions a customer is to make in the next time period, given the R(age at last purchase) and F (the number of repeat transactions made).

```
plot_frequency_recency_matrix(bgf)
```

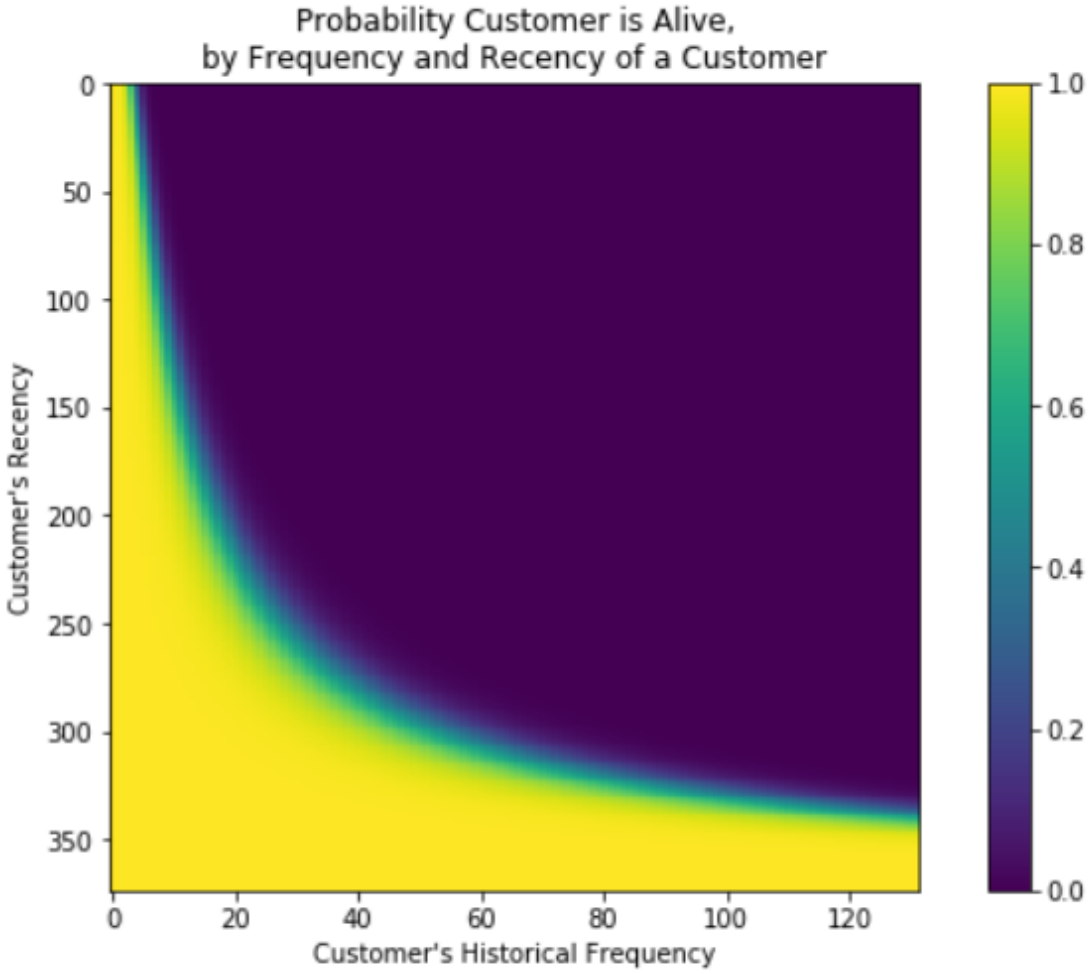


The RF plots maps a customer's expected purchases by the next year and probability that they're alive given the frequency / recency. Intuitively, we can see that customers with high frequency and recency are expected to purchase more in the future and have a higher chance of being alive. Customers in the white zone are of interest as well since they are 50/50 on leaving the company but we can still expect them to purchase about 2 to 2.5 times during the next year. These are the customers that may need a little customer servicing to come back and buy more. It is interesting to note that for a fixed recency, customer's with more frequency are more likely to be considered dead. This is a property of the model that illustrates a clear behavioral story:

> A customer making more frequent purchases is more likely to die off if we observe a longer period of inactivity than the customers previous intervals.

We can see that if a customer has bought 120 times and their latest purchase (R) was when they were 120 days back, then they are you best customer (bottom-right). The coldest customers are those that in the top-right corner: they bought a lot quickly, and haven't seen them in weeks. The tail around (20,50) represents the customers who buy infrequently, but are not seen recently, so they might buy again — we're unsure if they are dead or just between purchases. However, we can predict which customers are alive from below R/F plot:

```
plot_probability_alive_matrix(bgf)
```



Customers who have purchased recently are almost surely “alive”. Customers who have purchased a lot but not recently, are likely to have dropped out. And the more they bought in the past, the more likely they have dropped out. They are represented in the upper-right.

Ranking customers from best to worst

Let’s rank the customers from “highest expected purchases in the next period” to lowest. Models expose a method that will predict a customer’s expected purchases in the next period using their history.

```
t = 31*3
df['predicted_purchases'] =
bgf.conditional_expected_number_of_purchases_up_to_time(t,
df['frequency'], df['recency'], df['T'])
df.sort_values(by='predicted_purchases').tail(10)
```

CustomerID	frequency	recency	T	monetary_value	predicted_purchases
16422.0	47.0	352.0	369.0	702.472340	10.149887
13798.0	52.0	371.0	372.0	706.650962	11.138794
14527.0	53.0	367.0	369.0	155.016415	11.427290
13089.0	65.0	367.0	369.0	893.714308	13.974943
12971.0	70.0	369.0	372.0	159.211286	14.934019
14606.0	88.0	372.0	373.0	135.890114	18.687479
15311.0	89.0	373.0	373.0	677.729438	18.898020
17841.0	111.0	372.0	373.0	364.452162	23.526350
12748.0	113.0	373.0	373.0	298.360885	23.947326
14911.0	131.0	372.0	373.0	1093.661679	27.734064

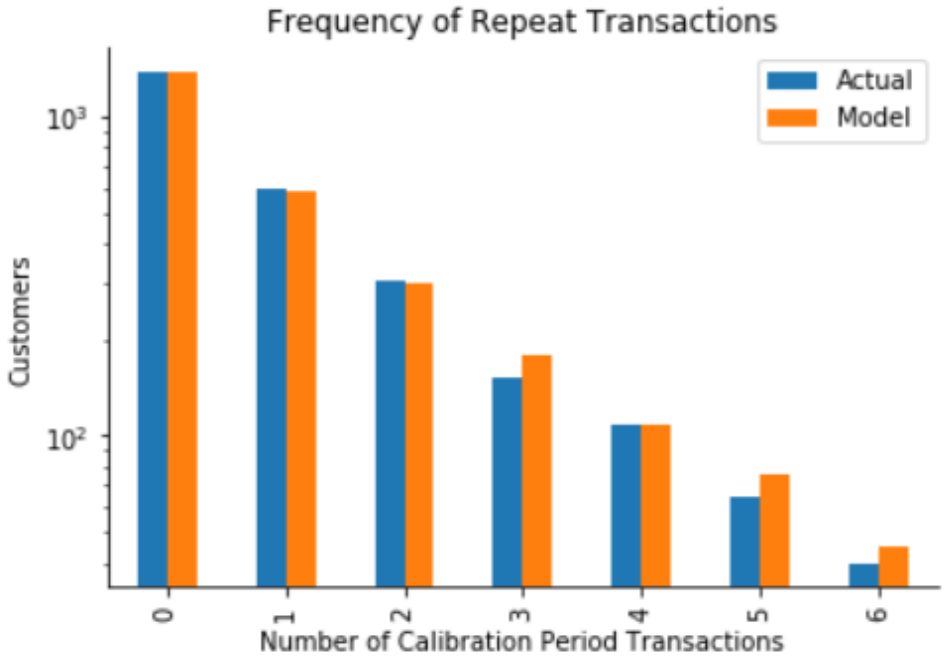
Listed here is our top 10 customers that the model expects them to purchase in the next 3 months. We can see that the customer who has made 131 purchases, and bought very recently, is probably going to buy again in the next period. The predicted_purchases

column displays their expected number of purchases while the other three columns represent their current RFM metrics. The BG/NBD model believes these individuals will be making more purchases within the near future as they are the current best customers.

Assessing model fit (Incorporating Heterogeneity)

After fitting the model, we're interested in seeing how well it is able to relate to our data. The first is to compare your data versus artificial data simulated with your fitted model's parameters.

```
plot_period_transactions(bgf)
```



The expected number of customers that are going to repeat purchase 0, 1, 2, 3 … 6 times in the future. For each number of repeat purchases (x-axis), we plot both what the model predicted and what the actual numbers were. As we can see, little to no errors in the fit for up to 6 repeat purchases. Let's do the next fact check.

However, it is always a good idea to compute the overall % error which is(predicted transactions/actual transactions -1) and the % error per transactions done in the calibration period. This will help us to quantify how close to reality the model is.

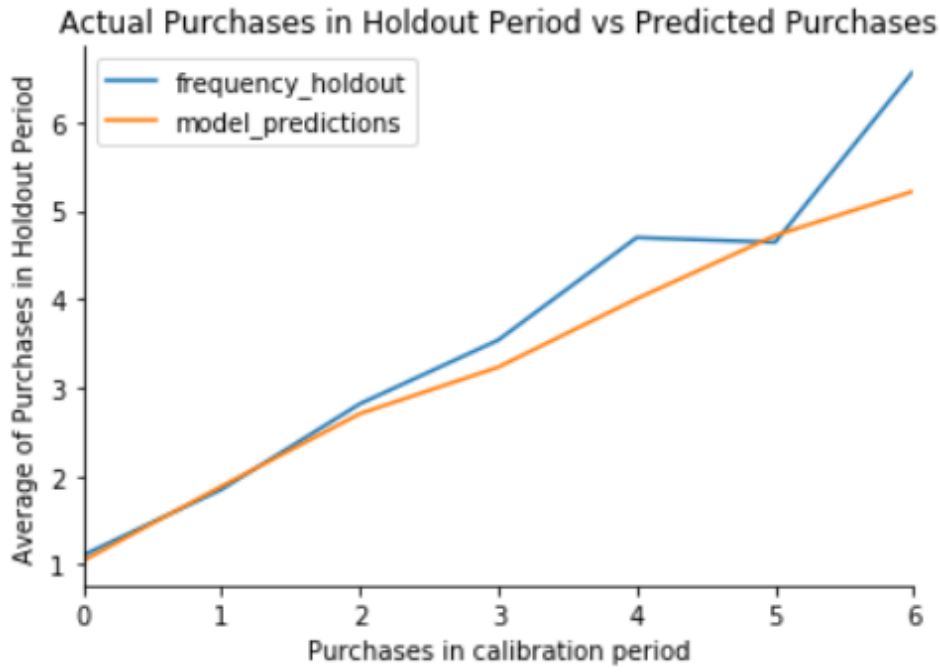
Model testing

```
summary_cal_holdout = calibration_and_holdout_data(data,
'CustomerID', 'InvoiceDate',
 calibration_period_end='2011-06-08',
 observation_period_end='2011-12-9' )
print(summary_cal_holdout.head())
```

```
            frequency_cal  recency_cal  ...  frequency_holdout  duration_holdout
CustomerID                              ...
12346.0               0.0          0.0  ...                0.0               184
12347.0               2.0        121.0  ...                4.0               184
12348.0               2.0        110.0  ...                1.0               184
12350.0               0.0          0.0  ...                0.0               184
12352.0               3.0         34.0  ...                3.0               184

[5 rows x 5 columns]
```

```
bgf.fit(summary_cal_holdout['frequency_cal'],
summary_cal_holdout['recency_cal'], summary_cal_holdout['T_cal'])
plot_calibration_purchases_vs_holdout_purchases(bgf,
summary_cal_holdout)
```



In this plot, we separate the data into both a in-sample (calibration) and validation (holdout) period. The sample period consists from the beginning to 2011–06–08; the validation period spans the rest of the duration (2011–06–09 to 2011–12–09). The plot 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). The plot 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). The orange and blue line presents the model prediction and actual result of the y-axis respectively. The model is able to accurately predict the customer base's behavior out of the sample, the model under-estimates at 4 purchases and after 5 purchases.

Customer transactions predictions

Based on customer history, we can now predict what an individual's future purchases might look like:

```
t = 10
individual = df.loc[12347]
bgf.predict(t, individual['frequency'], individual['recency'],
individual['T'])
```

0.15727742663038222

The model predicts that customer's (id:12347) future transaction is 0.157 in 10 days.

Conclusion

Customers represent the most important assets of a firm. Customer life-time value (CLTV) allows assessing their current and future value in a customer base. The CRM strategy and marketing resource allocation are based on this metric. Stakeholders therefore not only need to predict the retention but also the analyze the purchase behavior of their customers. Empirical investigation suggests that BG/NBD model arises by making a small, relatively inconsequential, change to the Pareto/NBD assumptions. The transition from an exponential distribution to a geometric process (to capture customer dropout) does not require any different psychological theories nor does it have any noteworthy managerial implications.

I can be reached at *here* .

Reference:

(1) Fader, P. S., Hardie, B. G., & Lee, K. L. (2005). "Counting your customers" the easy way: An alternative to the Pareto/NBD model. Marketing science, 24(2), 275–284.

(2)Schmittlein, David C., Donald G. Morrison, and Richard Colombo (1987), "Counting Your Customers: Who They Are and What Will They Do Next?" Management Science, 33 (January), 1–24.

Data Science | Customer Lifetime Value | Analytics | Data Modeling | Data Analysis