part3-clf

December 13, 2019

1 Part 3: Customer Lifetime Value Prediction

One of the most important metric we should estimate and tract is the Customer lifetime value.

Company invests in customers (acquisition costs, offline ads, promotions, discounts & etc.) to generate revenue and be profitable. Naturally, these actions make some customers super valuable in terms of lifetime value but there are always some customers who pull down the profitability. We need to identify these behavior patterns, segment customers and act accordingly.

To calculate lifetime value we need to select a time window. It can be anything like 3, 6, 12, 24 months. By the equation below, we can calculate lifetime value for each customer in that specific time window:

> ***Lifetime Value****: Total Gross Revenue - Total Cost*

This equation gives us the historical lifetime value. If we see some customers having very high negative lifetime value historically, it could be too late to take an action. At this point, we need to predict the future with machine learning.

Let's identify our path to predicting lifetime value: * Define an appropriate time frame for Customer Lifetime Value calculation * Identify the features we are going to use to predict future and create them * Calculate lifetime value (LTV) for training the machine learning model * Build and run the machine learning model * Check if the model is useful

1.1 Load the dataset

```
  InvoiceNo StockCode                          Description  Quantity  \
0    536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER         6
1    536365     71053                  WHITE METAL LANTERN         6
2    536365    84406B       CREAM CUPID HEARTS COAT HANGER         8
3    536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE         6
4    536365    84029E       RED WOOLLY HOTTIE WHITE HEART.         6

          InvoiceDate  UnitPrice  CustomerID         Country
0 2010-12-01 08:26:00       2.55     17850.0  United Kingdom
1 2010-12-01 08:26:00       3.39     17850.0  United Kingdom
2 2010-12-01 08:26:00       2.75     17850.0  United Kingdom
3 2010-12-01 08:26:00       3.39     17850.0  United Kingdom
4 2010-12-01 08:26:00       3.39     17850.0  United Kingdom
```

```
(3951,)
```

1.2 Lifetime value prediction

1.2.1 Time Frame

Deciding the time frame really depends on your industry, business model, strategy and more. For some industries, 1 year is a very long period while for the others it is very short. In our example, we will go ahead with **6 months**.

1.2.2 Features

RFM scores for each customer ID (which we calculated in the previous article) are the perfect candidates for feature set.

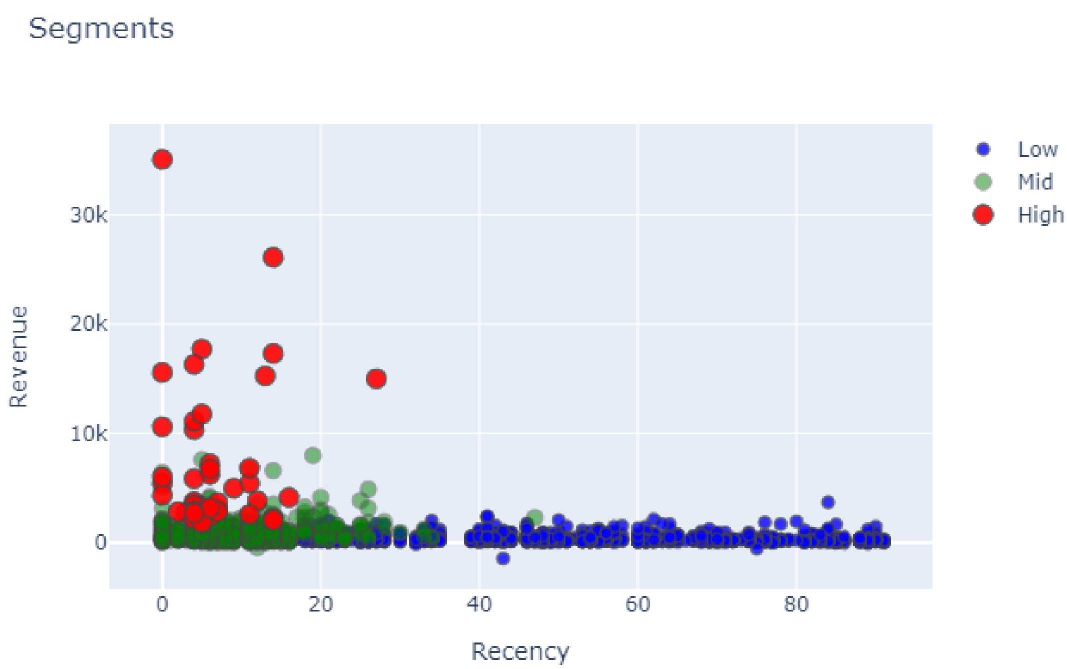
To implement it correctly, we need to split our dataset. We will take 3 months of data, calculate RFM and use it for predicting next 6 months. So we need to create two dataframes first and append RFM scores to them.

```
count                  541909
unique                  23260
top       2011-10-31 14:41:00
freq                     1114
first     2010-12-01 08:26:00
last      2011-12-09 12:50:00
Name: InvoiceDate, dtype: object

count                   95193
unique                   4852
top       2011-04-18 13:13:00
freq                      333
first     2011-03-01 08:30:00
last      2011-05-31 15:53:00
Name: InvoiceDate, dtype: object

count                  278966
unique                  11477
top       2011-10-31 14:41:00
freq                     1114
first     2011-06-01 07:37:00
last      2011-11-30 17:42:00
Name: InvoiceDate, dtype: object

   CustomerID  Recency  RecencyCluster  Frequency  FrequencyCluster  Revenue  \
0     14620.0       12               3         30                 0   393.28
1     15194.0        6               3         64                 0  1439.02
2     18044.0        5               3         57                 0   808.96
3     18075.0       12               3         35                 0   638.12
```

```
4     15241.0         0                3         64                 0   947.55

   RevenueCluster  OverallScore    Segment
0               0             3  Mid-Value
1               0             3  Mid-Value
2               0             3  Mid-Value
3               0             3  Mid-Value
4               0             3  Mid-Value
```

We have created our RFM scoring and now our feature set looks like above. Check customer scores on Recency, Frequency and Revenue accross Segments.

```
               Recency   Frequency      Revenue
Segment
Low-Value    49.650767   25.112436   394.738365
Mid-Value     9.812601   50.442649   910.635058
High-Value    6.468085  233.553191  7066.589149
```

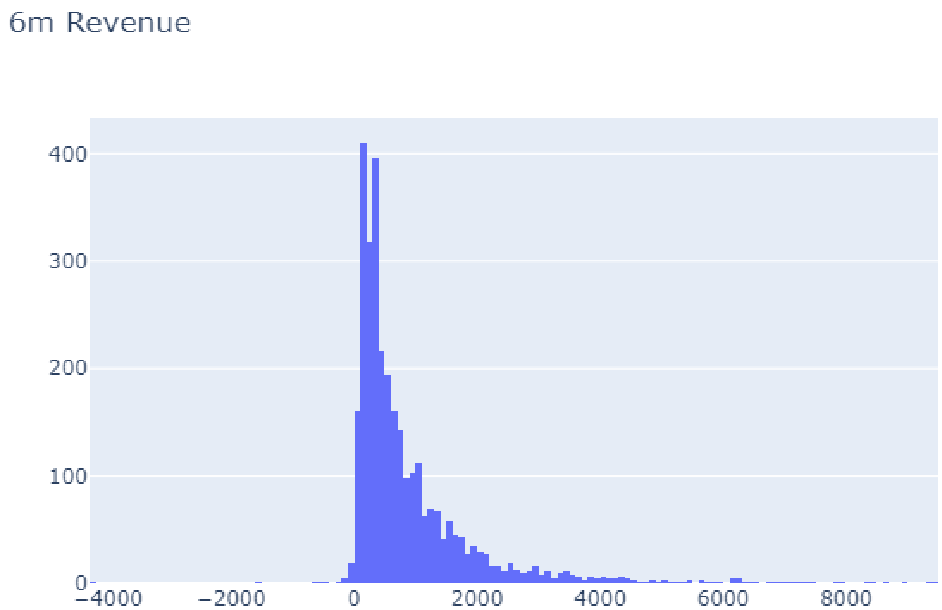


Calculate lifetime value (LTV) Since our feature set is ready, let's calculate 6 months LTV for each customer which we are going to use for training our model. There is no cost specified in the dataset. That's why Revenue becomes our LTV directly.

```
  InvoiceNo StockCode                   Description  Quantity  \
0    555156     23299  FOOD COVER WITH BEADS SET 2          6
1    555156     22847   BREAD BIN DINER STYLE IVORY         1
2    555157     23075     PARLOUR CERAMIC WALL HOOK        16
3    555157    47590B   PINK HAPPY BIRTHDAY BUNTING         6
4    555157     22423      REGENCY CAKESTAND 3 TIER         4

          InvoiceDate  UnitPrice  CustomerID         Country
0 2011-06-01 07:37:00       3.75     15643.0  United Kingdom
1 2011-06-01 07:37:00      16.95     15643.0  United Kingdom
2 2011-06-01 07:38:00       4.15     15643.0  United Kingdom
3 2011-06-01 07:38:00       5.45     15643.0  United Kingdom
4 2011-06-01 07:38:00      12.75     15643.0  United Kingdom
```

```
   CustomerID  m6_Revenue
0     12747.0     1666.11
1     12748.0    18679.01
2     12749.0     2323.04
3     12820.0      561.53
4     12822.0      918.98
```

```
count      3167.000000
mean       1239.685078
std        4782.390775
min       -4287.630000
25%         257.780000
50%         521.200000
75%        1148.670000
max      180469.050000
Name: m6_Revenue, dtype: float64
```

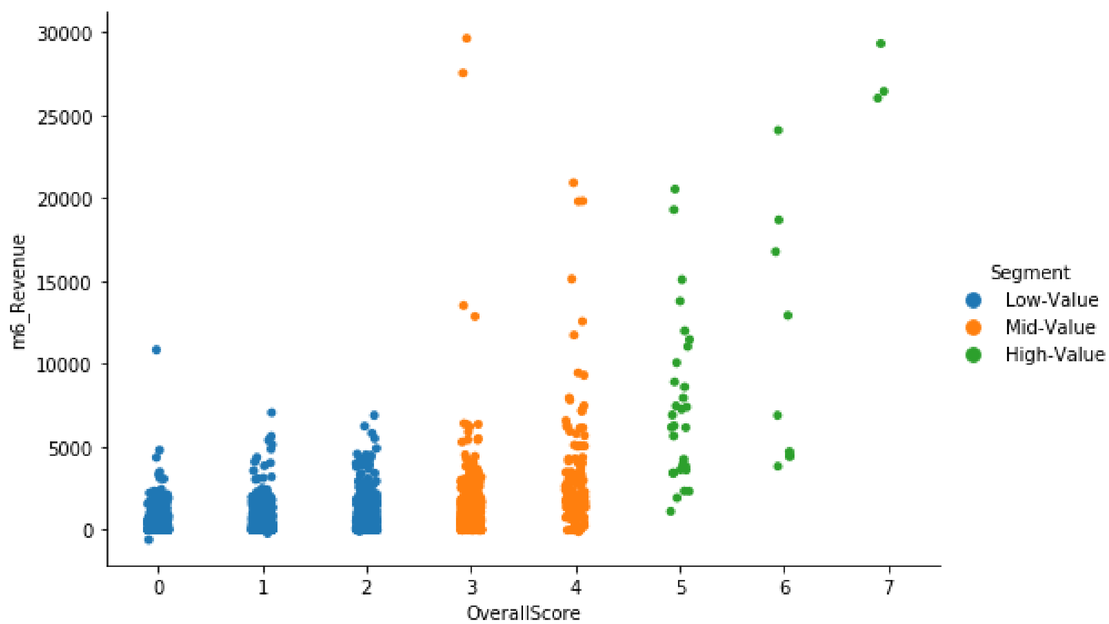


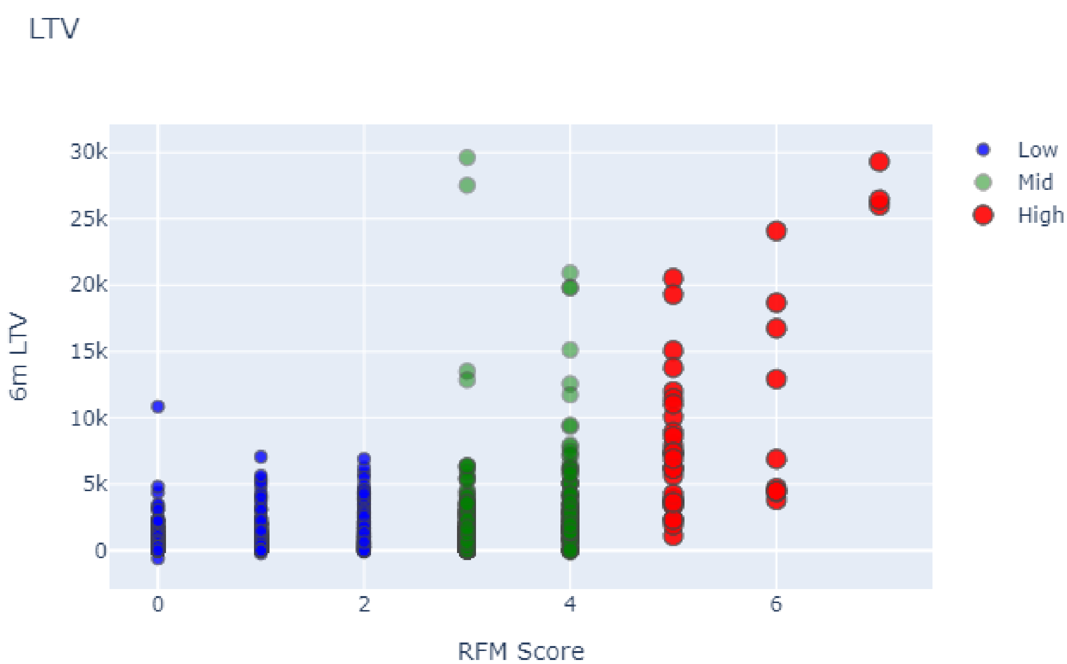
Histogram clearly shows we have customers with negative LTV. We have some outliers too. Filtering out the outliers makes sense to have a proper machine learning model.

Next step is to merge our 3 months and 6 months dataframes to see correlations between LTV and the feature set we have.

```
   CustomerID  Recency  RecencyCluster  Frequency  FrequencyCluster  Revenue  \
0     14620.0       12               3         30                 0   393.28
1     15194.0        6               3         64                 0  1439.02
2     18044.0        5               3         57                 0   808.96
3     18075.0       12               3         35                 0   638.12
4     15241.0        0               3         64                 0   947.55

   RevenueCluster  OverallScore    Segment  m6_Revenue
0               0             3  Mid-Value         NaN
1               0             3  Mid-Value     3232.20
2               0             3  Mid-Value      991.54
3               0             3  Mid-Value     1322.75
4               0             3  Mid-Value      791.04

Segment
High-Value    17658.143830
Low-Value       703.559983
```

```
Mid-Value      1715.514913
Name: m6_Revenue, dtype: float64

<seaborn.axisgrid.FacetGrid at 0x1beebfdd4e0>
```





Positive correlation is quite visible here. High RFM score means high LTV.

LTV Segments Before building the machine learning model, we need to identify what is the type of this machine learning problem. LTV itself is a regression problem. A machine learning model can predict the `$value` of the LTV. But here, we want *LTV segments*. Because it makes it more actionable and easy to communicate with other people. By applying K-means clustering, we can identify our existing LTV groups and build segments on top of it.

Considering business part of this analysis, we need to treat customers differently based on their predicted LTV. For this example, we will apply clustering and have 3 segments (number of segments really depends on your business dynamics and goals): * Low LTV * Mid LTV * High LTV

We are going to apply K-means clustering to decide segments and observe their characteristics:

```
   CustomerID  Recency  RecencyCluster  Frequency  FrequencyCluster  Revenue  \
0     14620.0       12               3         30                 0   393.28
1     15194.0        6               3         64                 0  1439.02
2     18044.0        5               3         57                 0   808.96
3     18075.0       12               3         35                 0   638.12
4     15241.0        0               3         64                 0   947.55

   RevenueCluster  OverallScore   Segment  m6_Revenue  LTVCluster
0               0             3  Mid-Value        0.00           1
1               0             3  Mid-Value     3232.20           0
2               0             3  Mid-Value      991.54           1
3               0             3  Mid-Value     1322.75           1
4               0             3  Mid-Value      791.04           1

             count         mean          std      min       25%       50%  \
LTVCluster
0           1394.0   396.137189   419.891843  -609.40     0.000   294.220
1            371.0  2492.794933   937.341566  1445.31  1731.980  2162.930
2             56.0  8222.565893  2983.572030  5396.44  6151.435  6986.545

                  75%       max
LTVCluster
0            682.4300   1429.87
1           3041.9550   5287.39
2           9607.3225  16756.31
```

We have finished LTV clustering and now have 3 clusters where cluster 2 is the best with average 8.2k LTV whereas cluster 0 is the worst with 396.

Prepare data for machine learning There are few more steps before training the machine learning model: * We need to do some feature engineering. We should convert categorical columns to numerical columns. * We will check the correlation of features against our label, LTV clusters. * Split data into features and lables (X, y) and create training and test sets.

```
   CustomerID  Recency  RecencyCluster  Frequency  FrequencyCluster  Revenue  \
0     14620.0       12               3         30                 0   393.28
1     18044.0        5               3         57                 0   808.96
2     18075.0       12               3         35                 0   638.12
3     15241.0        0               3         64                 0   947.55
4     15660.0        4               3         34                 0   484.62

   RevenueCluster  OverallScore  m6_Revenue  LTVCluster  Segment_High-Value  \
0               0             3        0.00           0                   0
1               0             3      991.54           0                   0
2               0             3     1322.75           0                   0
3               0             3      791.04           0                   0
4               0             3      858.09           0                   0

   Segment_Low-Value  Segment_Mid-Value
0                  0                  1
1                  0                  1
2                  0                  1
3                  0                  1
4                  0                  1
```

```
LTVCluster            1.000000
Revenue               0.600491
RevenueCluster        0.463930
OverallScore          0.373231
FrequencyCluster      0.366366
Frequency             0.359601
Segment_High-Value    0.353218
RecencyCluster        0.236899
Segment_Mid-Value     0.166854
CustomerID           -0.028401
Recency              -0.237249
Segment_Low-Value    -0.266008
Name: LTVCluster, dtype: float64
```

We see that 3 features Revenue, OverallScore and Frequency will be helpful for our machine learning models.

Note: we cannot user m6_Revenue as it comes from dataset used to generate label.

Build and run the machine learning model Since we have the training and test sets we can build our model.

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bynode=1, colsample_bytree=1, gamma=0, learning_rate=0.1,
```

```
max_delta_step=0, max_depth=5, min_child_weight=1, missing=None,
n_estimators=100, n_jobs=-1, nthread=None,
objective='multi:softprob', random_state=0, reg_alpha=0,
reg_lambda=1, scale_pos_weight=1, seed=None, silent=None,
subsample=1, verbosity=1)
```

Check if the model is useful We used a quite strong XGBoost ML library to do the classification for us. It has become a multi classification model since we had 3 groups (clusters). Let's look at the initial results:

```
Accuracy of XGB classifier on training set: 0.92
Accuracy of XGB classifier on test set: 0.78
```

Accuracy shows 79% on the test set. Looks really good. Or does it?

```
LTVCluster
0    0.765513
1    0.203734
2    0.030752
Name: CustomerID, dtype: float64
```

First we need to check our benchmark. Biggest cluster we have is cluster 0 which is 76.5% of the total base. If we blindly say, every customer belongs to cluster 0, then our accuracy would be 76.5%.

79% vs 76.5% tell us that our machine learning model is mildly useful one but needs some improvement for sure. We should find out where the model is failing.

We can identify that by looking at classification report:

```
              precision    recall  f1-score   support

           0       0.84      0.91      0.88       140
           1       0.42      0.30      0.35        37
           2       0.60      0.50      0.55         6

   micro avg       0.78      0.78      0.78       183
   macro avg       0.62      0.57      0.59       183
weighted avg       0.75      0.78      0.76       183
```

Precision and recall are acceptable for 0. As an example, for cluster 0 (Low LTV), if model tells us this customer belongs to cluster 0, 85 out of 100 will be correct (precision). And the model successfully identifies 92% of actual cluster 0 customers (recall).

We really need to improve the model for other clusters. For example, we barely detect 46% of Mid LTV customers. Possible actions to improve those points: * Adding more features and improve feature engineering * Try different models other than XGBoost * Apply hyper parameter tuning to current model * Add more data to the model if possible

Great! Now we have a machine learning model which predicts the future LTV segments of our customers. We can easily adapt our actions based on that. For example, we definitely do not want to lose customers with high LTV. So we will focus on Churn Prediction in Part 4.