[DATA DRIVEN GROWTH WITH PYTHON](https://towardsdatascience.com/tagged/data-driven-growth" \t "_blank)

**Customer Lifetime Value Prediction**

LTV prediction with XGBoost Multi-classification

[Barış Karaman](https://towardsdatascience.com/@karamanbk?source=post_page-----6017802f2e0f----------------------)

[Follow](https://medium.com/m/signin?operation=register&redirect=https%3A%2F%2Ftowardsdatascience.com%2Fdata-driven-growth-with-python-part-3-customer-lifetime-value-prediction-6017802f2e0f&source=-d48af1458440-------------------------follow_byline-)

[May 4](https://towardsdatascience.com/data-driven-growth-with-python-part-3-customer-lifetime-value-prediction-6017802f2e0f?source=post_page-----6017802f2e0f----------------------) · 7 min read

<https://towardsdatascience.com/data-driven-growth-with-python-part-3-customer-lifetime-value-prediction-6017802f2e0f>

**Introduction**

This series of articles was designed to explain how to use Python in a simplistic way to fuel your company’s growth by applying the predictive approach to all your actions. It will be a combination of programming, data analysis, and machine learning.

I will cover all the topics in the following nine articles:

1- [Know Your Metrics](https://towardsdatascience.com/data-driven-growth-with-python-part-1-know-your-metrics-812781e66a5b?source=post_page---------------------------)

2- [Customer Segmentation](https://towardsdatascience.com/data-driven-growth-with-python-part-2-customer-segmentation-5c019d150444?source=post_page---------------------------)

3- **Customer Lifetime Value Prediction**

4- [Churn Prediction](https://towardsdatascience.com/churn-prediction-3a4a36c2129a?source=post_page---------------------------)

[5- Predicting Next Purchase Day](https://towardsdatascience.com/predicting-next-purchase-day-15fae5548027?source=post_page---------------------------)

[6- Predicting Sales](https://towardsdatascience.com/predicting-sales-611cb5a252de?source=post_page---------------------------)

[7- Market Response Models](https://towardsdatascience.com/market-response-models-baf9f9913298)

[8- Uplift Modeling](https://towardsdatascience.com/uplift-modeling-e38f96b1ef60)

[9- A/B Testing Design and Execution](https://towardsdatascience.com/a-b-testing-design-execution-6cf9e27c6559)

Articles will have their own code snippets to make you easily apply them. If you are super new to programming, you can have a good introduction for [Python](https://www.kaggle.com/learn/python?source=post_page---------------------------)and [Pandas](https://www.kaggle.com/learn/pandas?source=post_page---------------------------) (a famous library that we will use on everything) here. But still without a coding introduction, you can learn the concepts, how to use your data and start generating value out of it:

*Sometimes you gotta run before you can walk — Tony Stark*

As a pre-requisite, be sure J[upyter Notebook](https://jupyter.readthedocs.io/en/latest/install.html?source=post_page---------------------------) and P[ython](https://www.python.org/downloads/?source=post_page---------------------------) are installed on your computer. The code snippets will run on Jupyter Notebook only.

Alright, let’s start.

**Part 3: Customer Lifetime Value**

In the [previous article](https://medium.com/@karamanbk/data-driven-growth-with-python-part-2-customer-segmentation-5c019d150444), we segmented our customers and found out who are the best ones. Now it’s time to measure one of the most important metric we should closely track: **Customer Lifetime Value.**

We invest 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.

Calculating Lifetime Value is the easy part. First we need to select a time window. It can be anything like 3, 6, 12, 24 months. By the equation below, we can have Lifetime Value for each customer in that specific time window:

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

This equation now 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:

**We are going to build a simple machine learning model that predicts our customers lifetime value.**

**Lifetime Value Prediction**

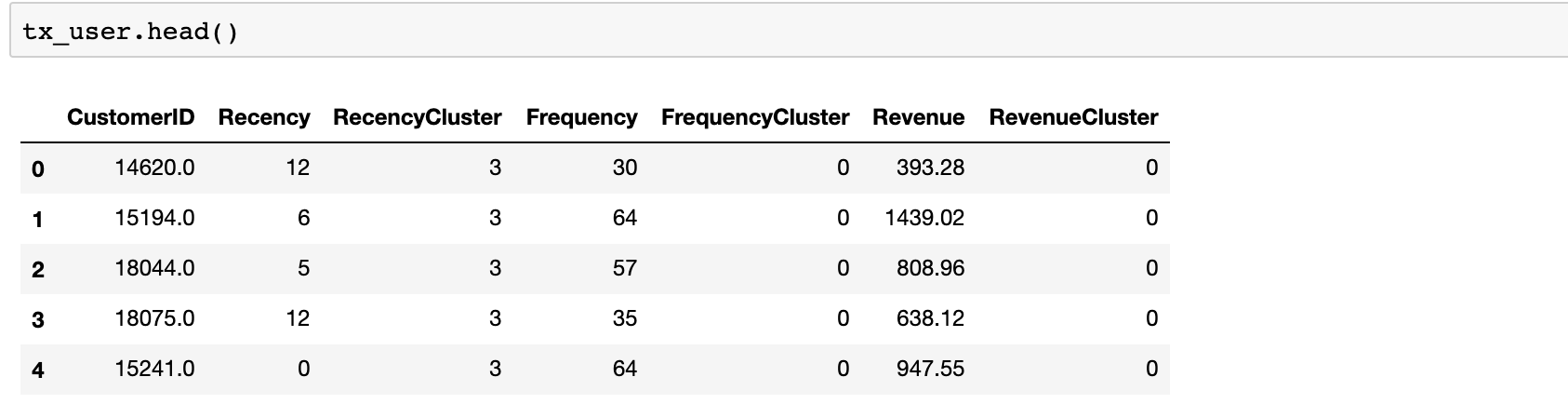
We will continue using our [online retail dataset](https://www.kaggle.com/vijayuv/onlineretail) for this example as well. Let’s identify our path to glory:

* 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

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.**

RFM scores for each customer ID (which we calculated in the [previous article](https://medium.com/@karamanbk/data-driven-growth-with-python-part-2-customer-segmentation-5c019d150444)) 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.

We have created our RFM scoring and now our feature set looks like below:

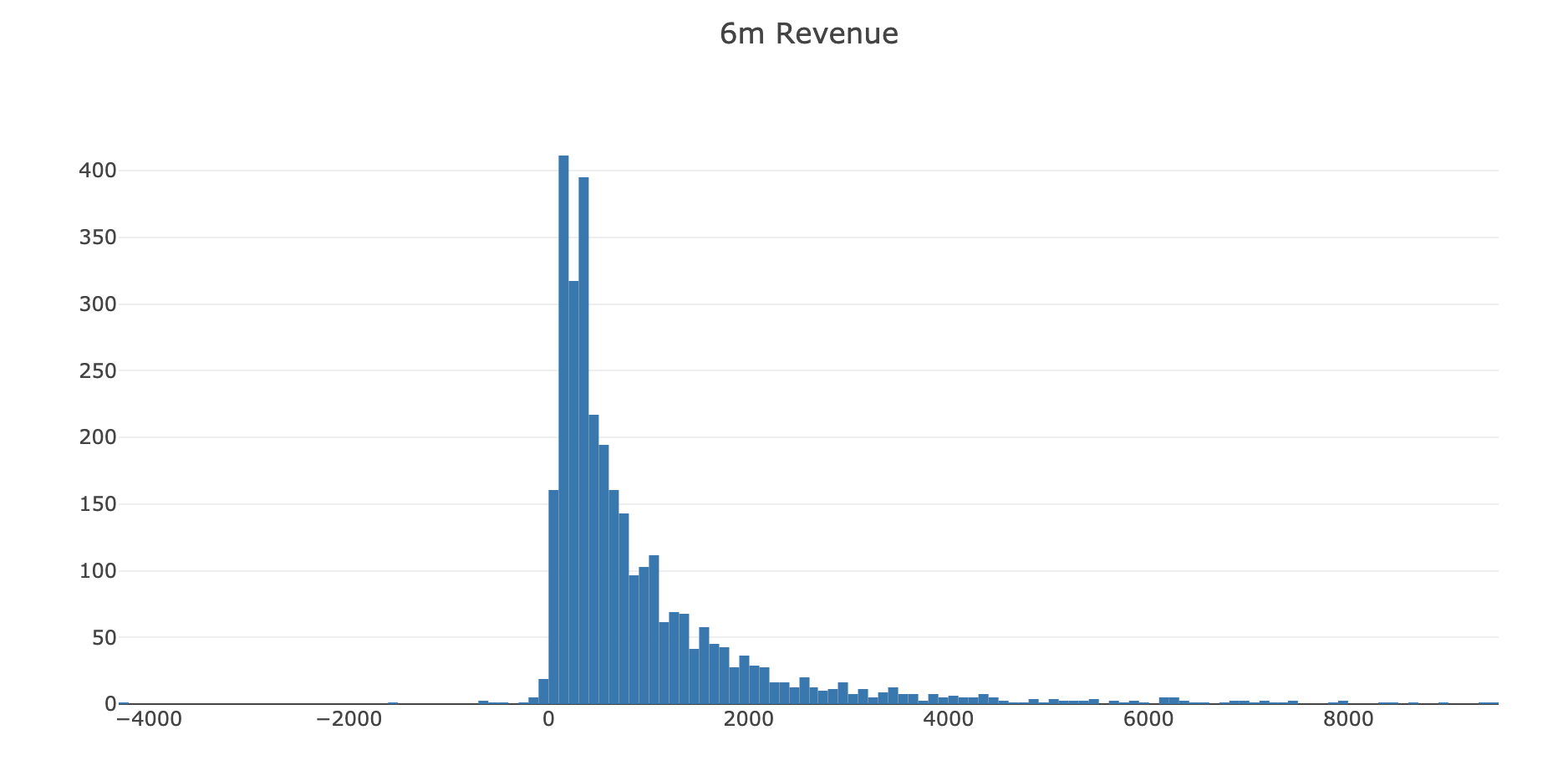


I won’t go over the details of RFM scoring as I would be repeating Part 2.

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.

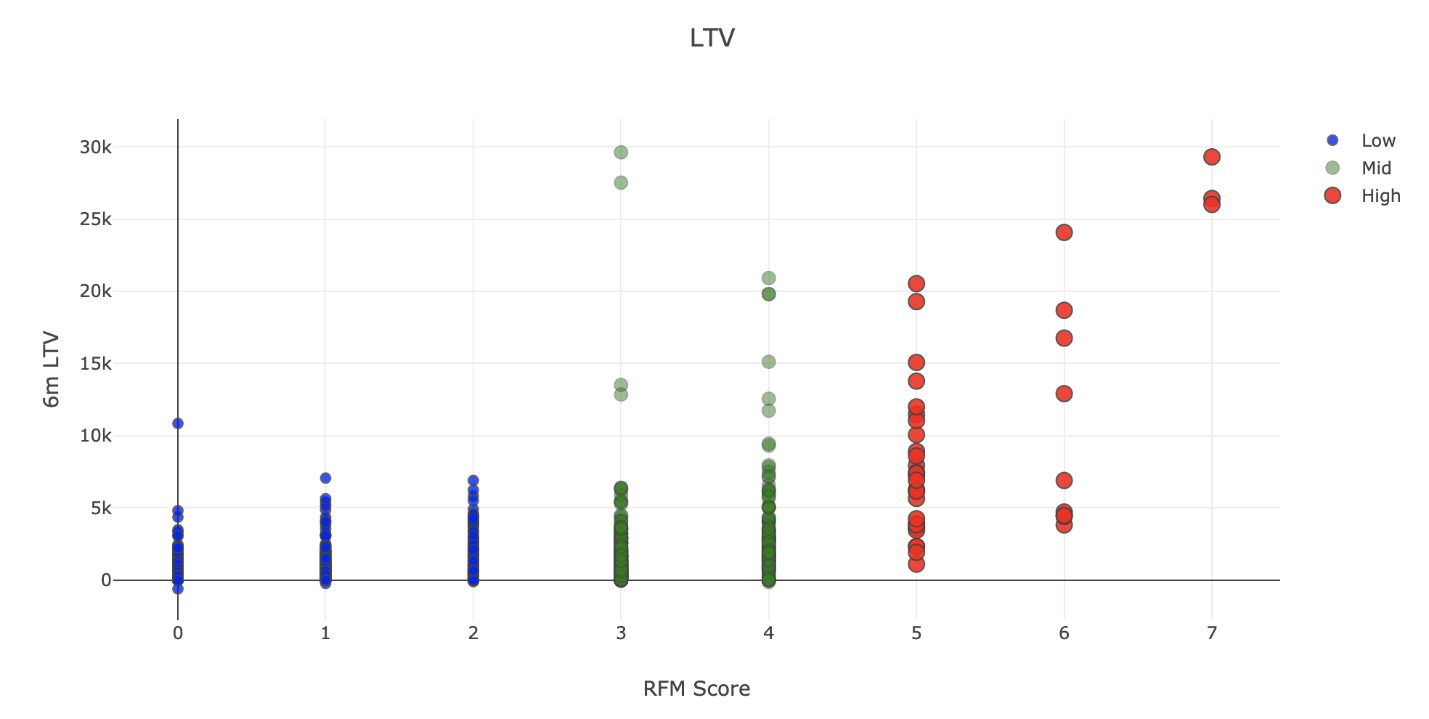
This code snippet calculates the LTV and plot its histogram:



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.

Ok, next step. We will merge our 3 months and 6 months dataframes to see correlations between LTV and the feature set we have.

The code below merges our feature set and LTV data and plots LTV vs overall RFM score:



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

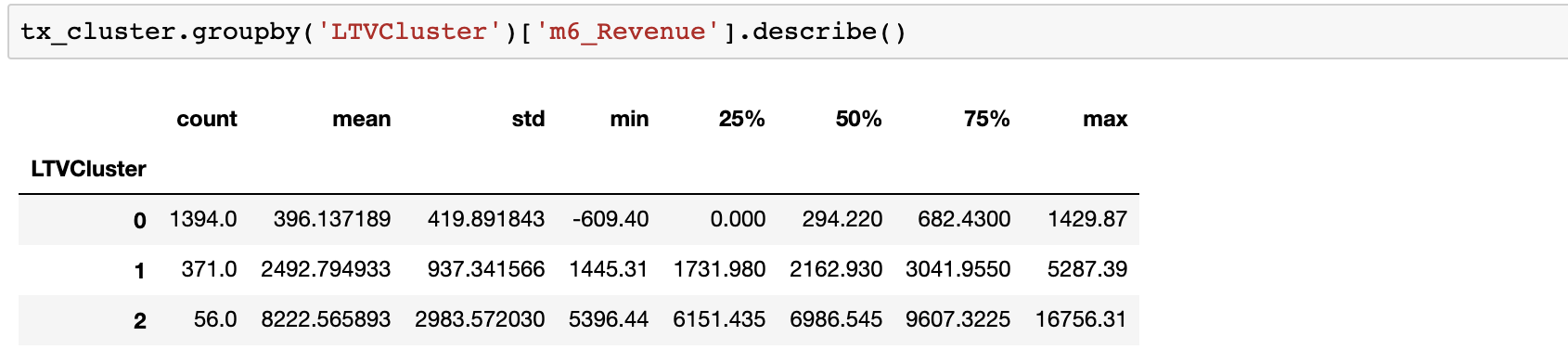
Before building the machine learning model, we need to identify what is the type of this machine learning problem. LTV itself is a regression problem. A machine learning model can predict the $ value of the LTV. But here, we want LTV segments. Because it makes it more actionable and easy to communicate 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:

We have finished LTV clustering and here are the characteristics of each clusters:



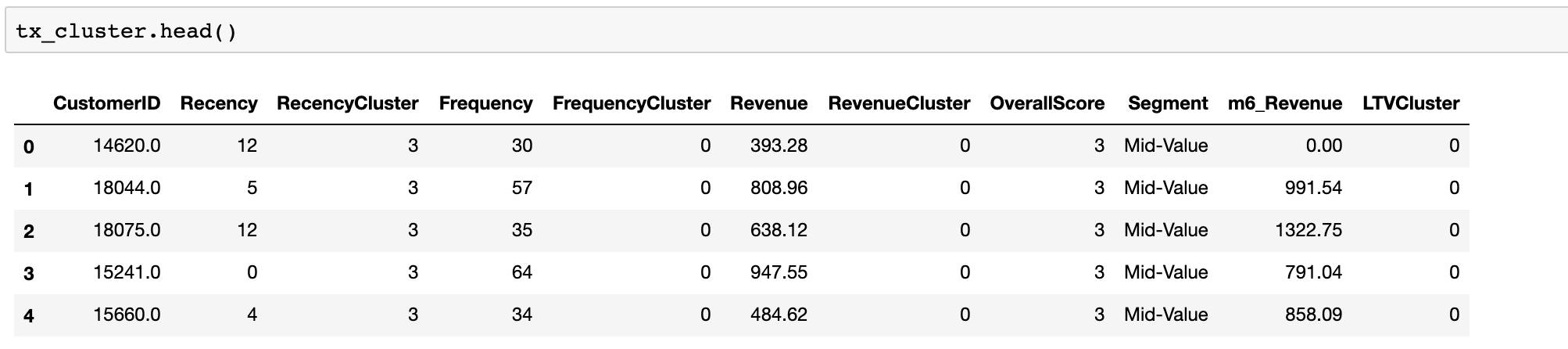
2 is the best with average 8.2k LTV whereas 0 is the worst with 396.

There are few more step before training the machine learning model:

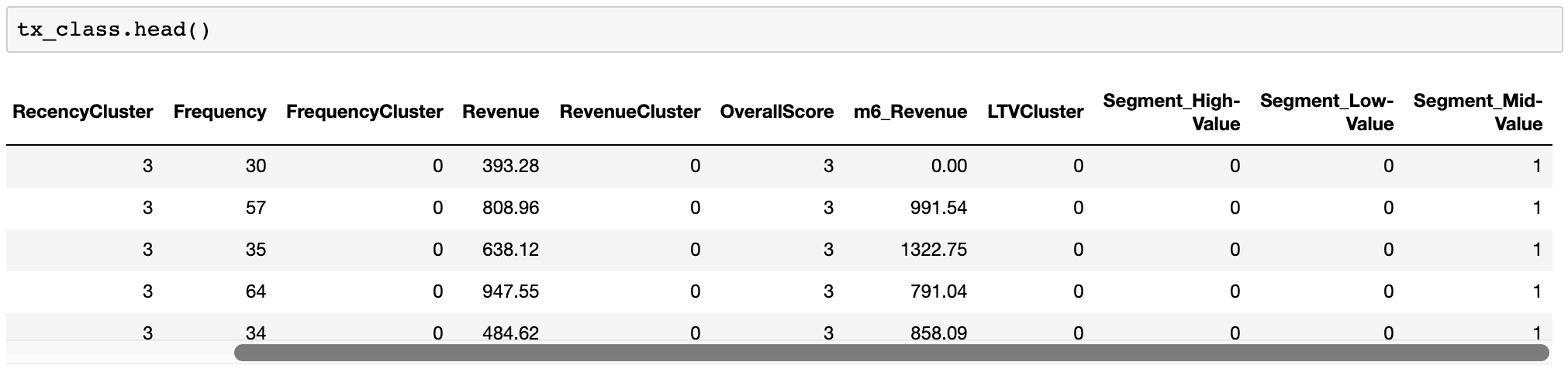
* Need to do some feature engineering. We should convert categorical columns to numerical columns.
* We will check the correlation of features against our label, LTV clusters.
* We will split our feature set and label (LTV) as X and y. We use X to predict y.
* Will create Training and Test dataset. Training set will be used for building the machine learning model. We will apply our model to Test set to see its real performance.

The code below does it all for us:

Let’s start with the first line. **get\_dummies()**method converts categorical columns to 0–1 notations. See what it exactly does with the example:



This was our dataset before get\_dummies(). We have one categorical column which is Segment. What happens after applying get\_dummies():



Segment column is gone but we have new numerical ones which represent it. We have converted it to 3 different columns with 0 and 1 and made it usable for our machine learning model.

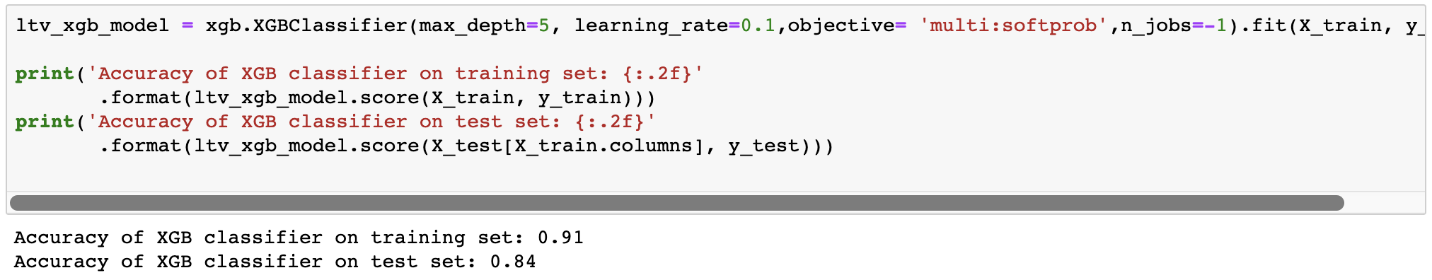
Lines related to correlation make us have the data below:



We see that 3 months Revenue, Frequency and RFM scores will be helpful for our machine learning models.

Since we have the training and test sets we can build our model.

We used a quite strong ML library called XGBoost 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:

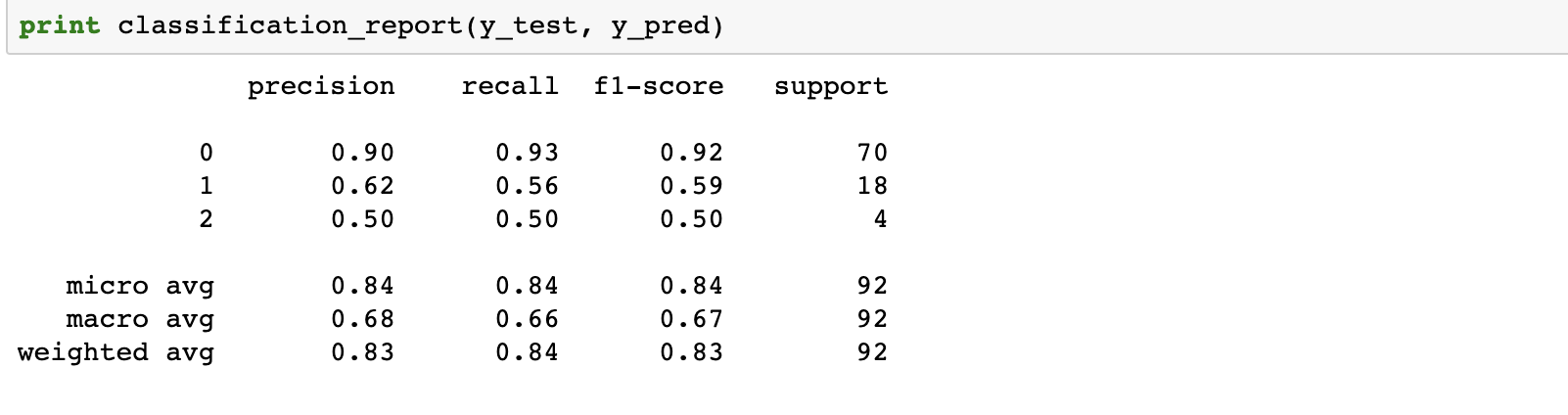


Accuracy shows 84% on the test set. Looks really good. Or does it?

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%.

84% vs 76.5% tell us that our machine learning model is a 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 and recall are acceptable for 0. As an example, for cluster 0 (Low LTV), if model tells us *this customer belongs to cluster 0*, 90 out of 100 will be correct (precision). And the model successfully identifies 93% of actual cluster 0 customers (recall). We really need to improve the model for other clusters. For example, we barely detect 56% 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](https://towardsdatascience.com/churn-prediction-3a4a36c2129a).

You can find the jupyter notebook for this part [here](https://gist.github.com/karamanbk/29983fdf5572a838f53163a6010b14f9).