Customer LifeTime Value Prediction

Sai Jyothi Gurram

Data Overview

- Brazilian Ecommerce dataset had various features tying an order to it’s product, customer, seller, marketing(MQL & lead characteristic data).
- After careful analysis, the marketing metrics had a lot of missing data and therefore haven't been used in this analysis.
- All the data sources were combined and the redundent features were removed.
- Finally a dataframe with order_id, customer_id, customer_unique_id, product_category, revenue, order_date was created.

Problem Statement

- Predict the Customer Revenue at transaction level.

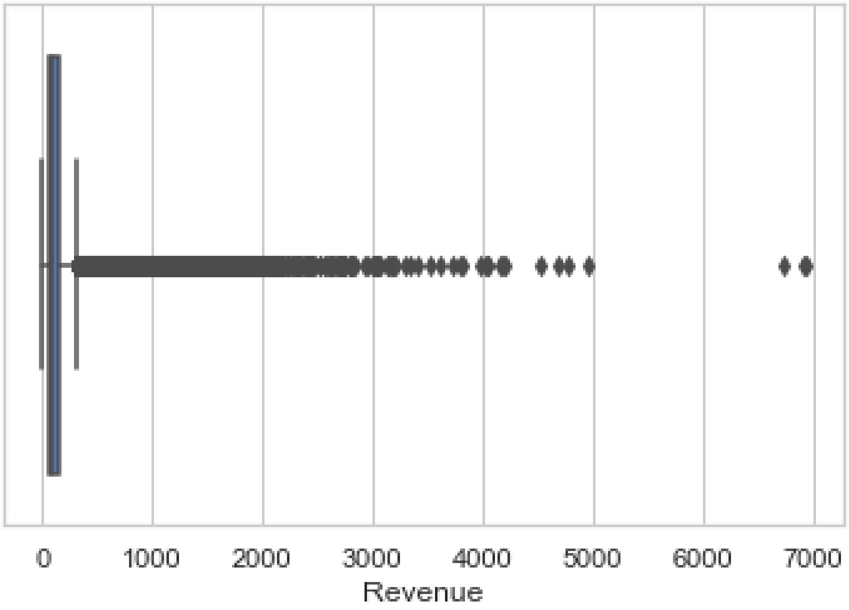
Null Values

```
Data columns (total 11 columns):
order_id                        112650 non-null object
order_item_id                   112650 non-null int64
product_id                      112650 non-null object
seller_id                       112650 non-null object
shipping_limit_date             112650 non-null datetime64[ns]
price                           112650 non-null float64
freight_value                   112650 non-null float64
product_category_name_english   111023 non-null object
customer_id                     112650 non-null object
order_purchase_timestamp        112650 non-null datetime64[ns]
customer_unique_id              112650 non-null object
```

- There weren’t any null values in most of the columns except product category
- Imputed missing product categories as Other

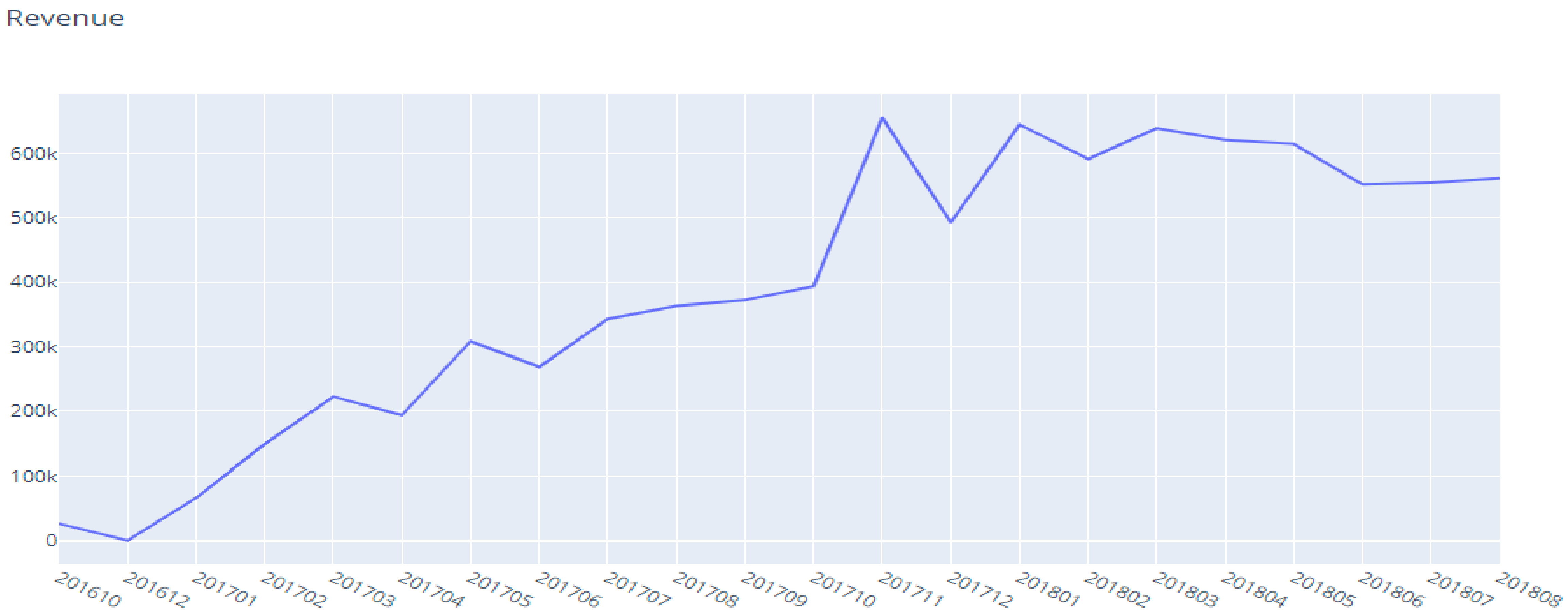
Outliers

```
count    112643.000000
mean        140.648172
std         190.729357
min           6.080000
25%          55.225000
50%          92.320000
75%         157.940000
max        6929.310000
Name: Revenue, dtype: float64
```



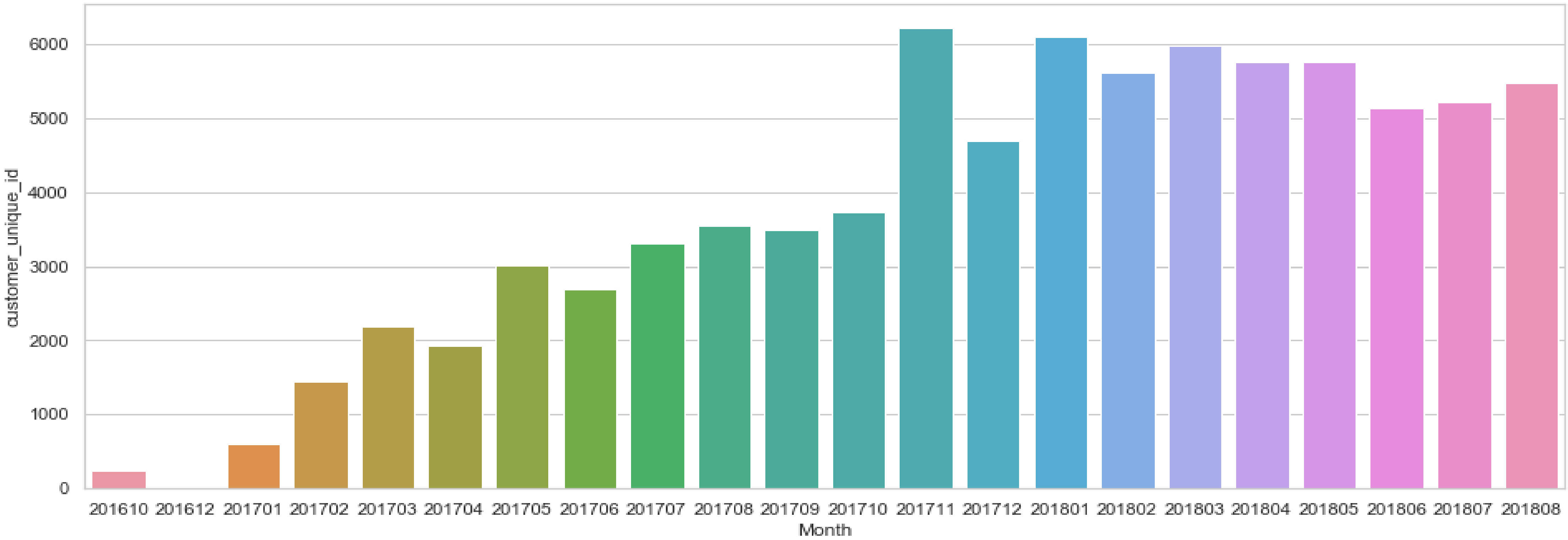
- Clearly there are outliers in the revenue based on the above results and boxplot. There are many values beyond the 4rth Quartile.
- Removed all the outliers that are not within 1.5*times InterQuartile Range (16674 rows).

Revenue Distribution - EDA



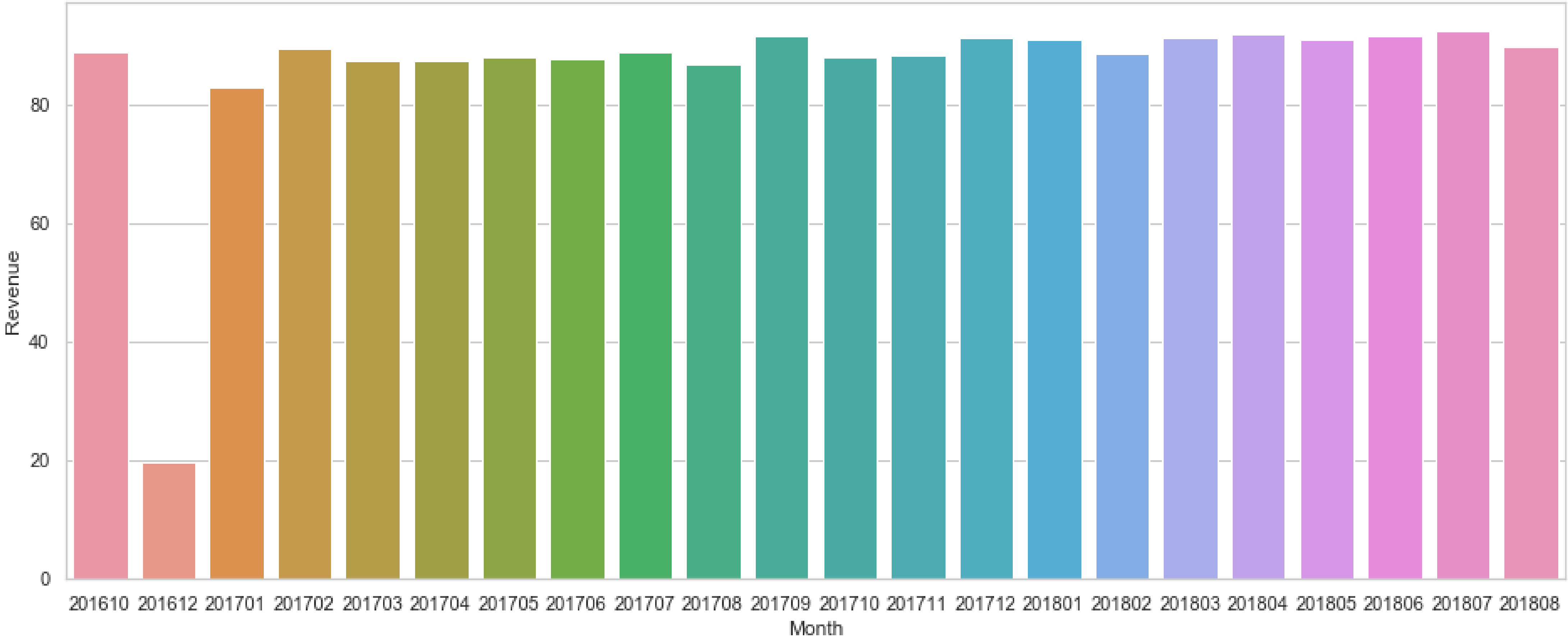
The Revenue distribution consistently increased with a peak in November 2017

Customer Distribution - EDA



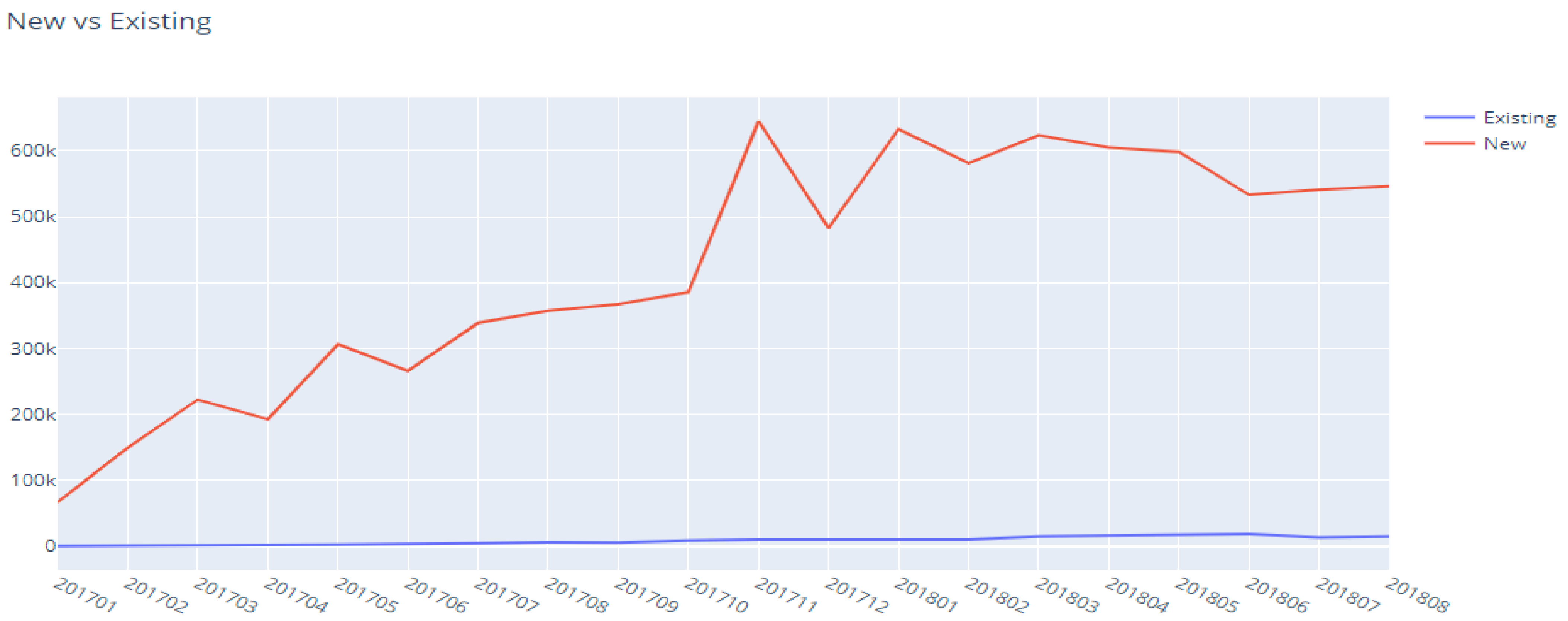
Similar to Revenue the number of unique customers also increased consistently.

Average Order Size -EDA



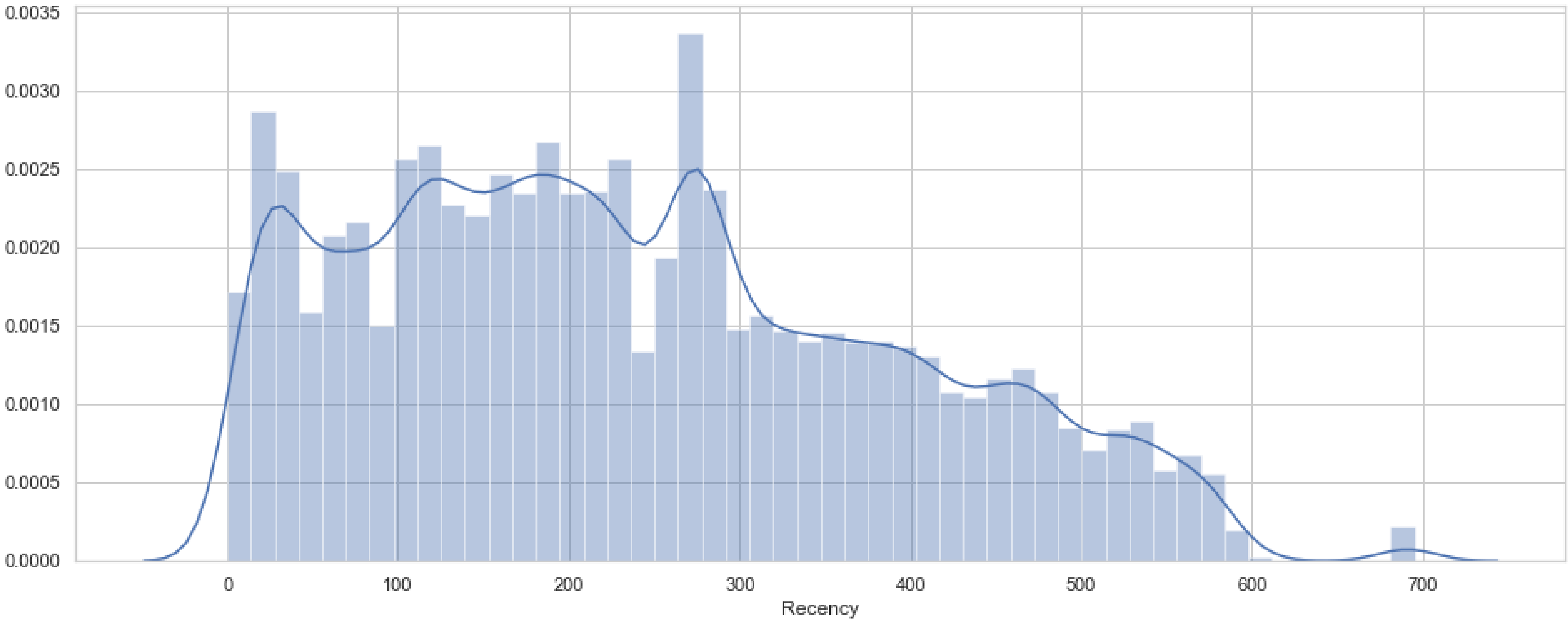
Average order is consistent across all months (except December 2016)

New Vs Existing Customers - EDA



- Created a new feature user_type to distinguish New customer vs existing customer.
- Existing customers Revenue contribution is negligible i.e organization wasn't able to retain existing customers.

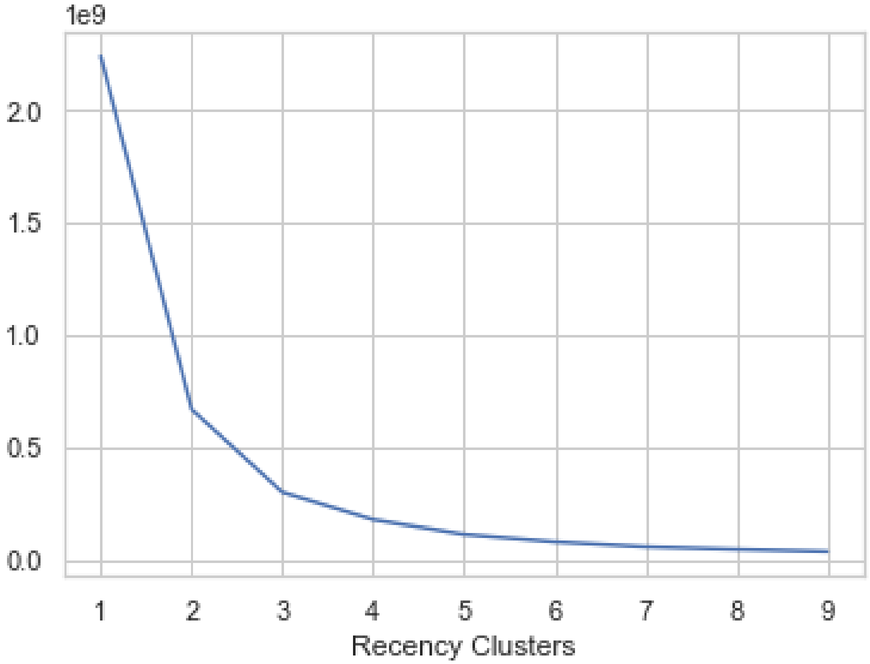
Recency Distribution - EDA



Created a new feature Recency – Difference between maximum order date and actual order date

More than 70% of customers were acquired within last year.

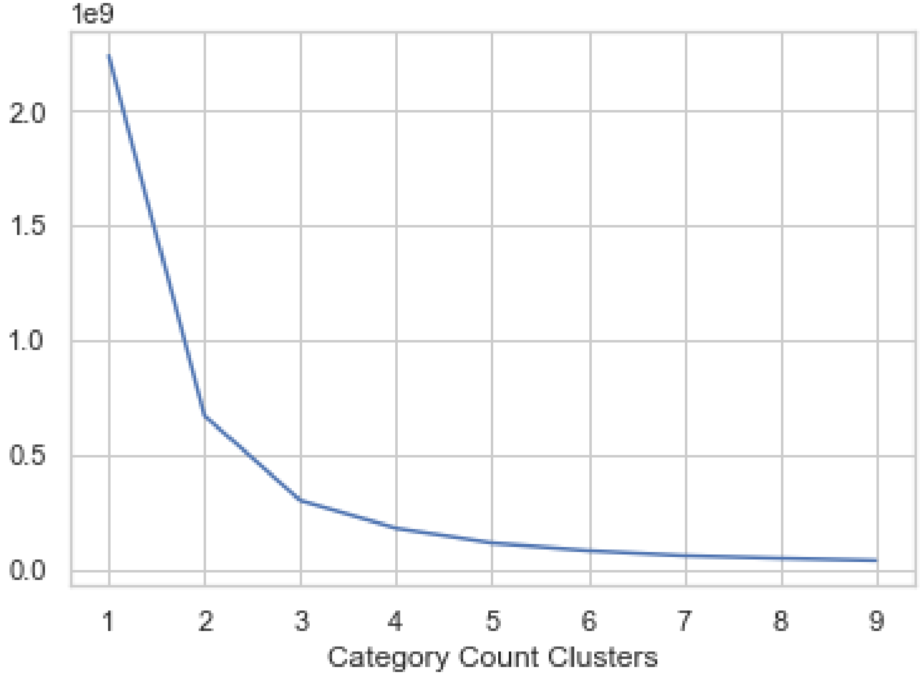
K-Means Clustering for Recency Feature



RecencyCluster	count	mean	std	min	25%	50%	75%	max
0	17254.0	484.022256	59.580103	399.0	436.0	475.0	527.0	695.0
1	27968.0	183.385333	34.977054	123.0	154.0	183.0	214.0	248.0
2	25484.0	61.972179	36.124201	0.0	28.0	62.0	97.0	122.0
3	25263.0	313.890749	43.341408	249.0	276.0	307.0	351.0	398.0

- Used Elbow Method to identify optimal k value(here 4)
- Divided Recency into 4 clusters
- Recency 0 - Very Old , Recency1 – Less Recent, Receny2- Recent, Recency3 - Old

K-Means Clustering for Product Category Count Feature



product_category	count	mean	std	min	25%	50%	75%	max
0	46422.0	7928.053681	1294.567788	6318.0	6882.0	7527.0	8059.0	10167.0
1	32292.0	3365.096247	662.552735	2341.0	2710.0	3473.0	4113.0	4233.0
2	17255.0	866.165807	625.054188	1.0	295.0	732.0	1467.0	1902.0

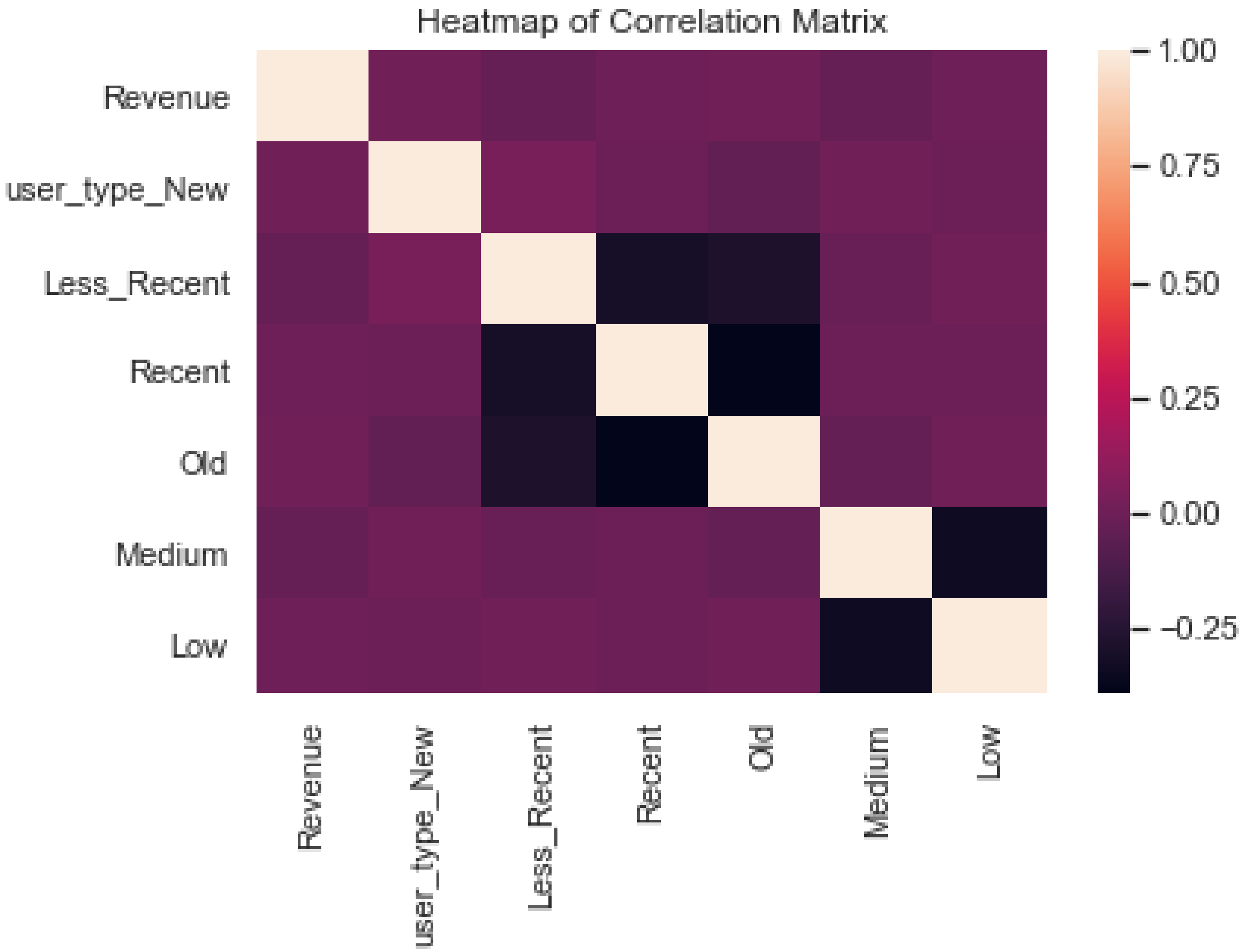
- There are 72 distinct product categories
- Created New feature category count (the count of various product categories)
- Used K-means elbow method and clustered into 3 categories
- Category0 – High, Category1 – Medium, Category2 - Low

Base Model

	Actual	model_1	model_2	Ensemble
48173	86.40	88.940870	0.494812	89.435682
12479	46.75	90.896047	-0.117988	90.778060
56644	116.94	88.940870	0.494812	89.435682
70179	25.75	88.940870	0.494812	89.435682
67632	64.03	91.355188	-0.117988	91.237200

- Created a Base decision tree model and predicted the revenue
- Created a second model by training the model on model1 errors
- Finally created an ensemble
- The ensemble now has lower error compared to model1

Correlation

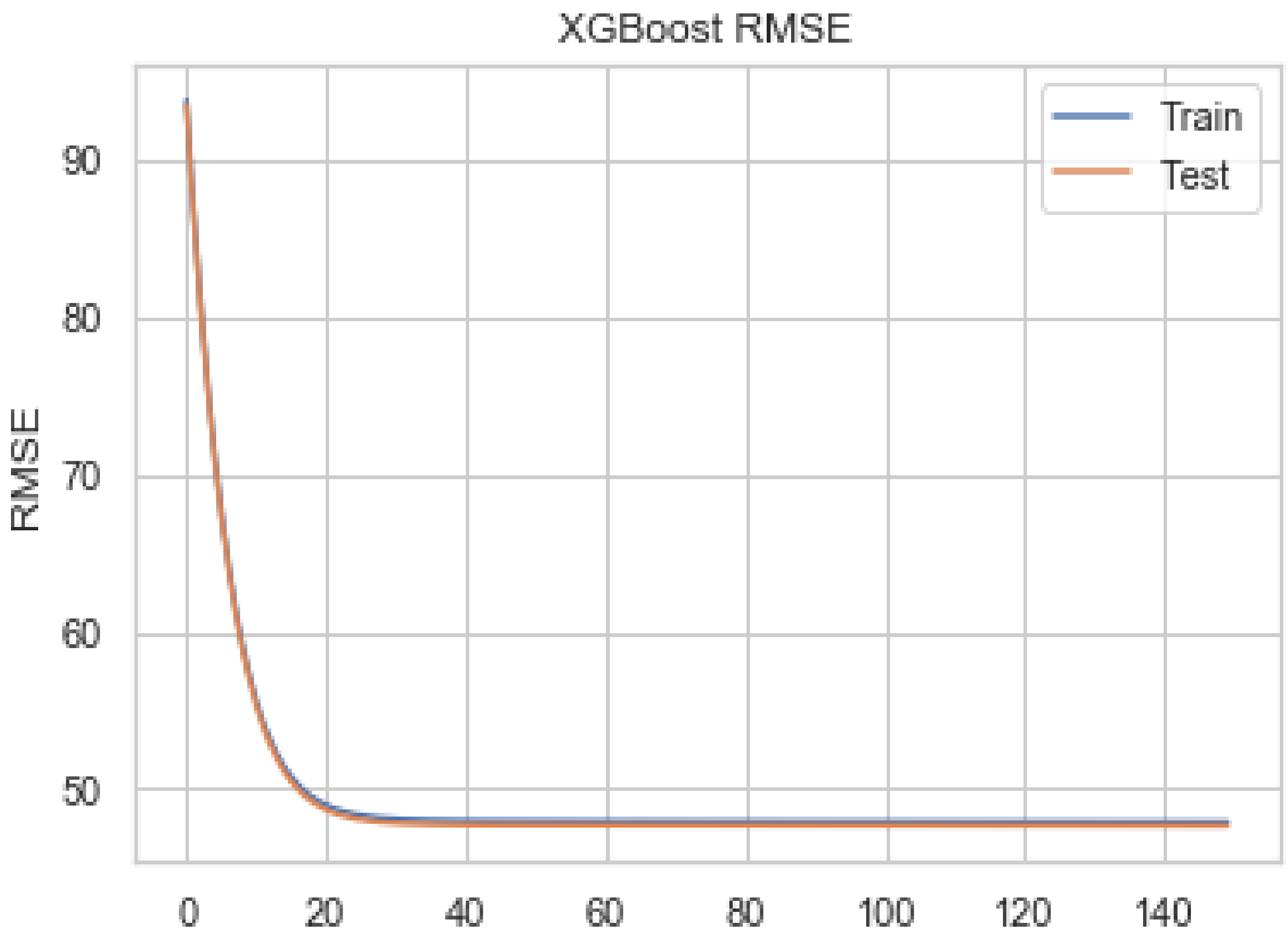


Model Selection

Tree Count	Average Error
1	62.747418
2	43.924619
3	30.748293
4	21.524089
5	15.067433
6	10.547623
7	7.383321
8	5.168529
9	3.618196
10	2.532704

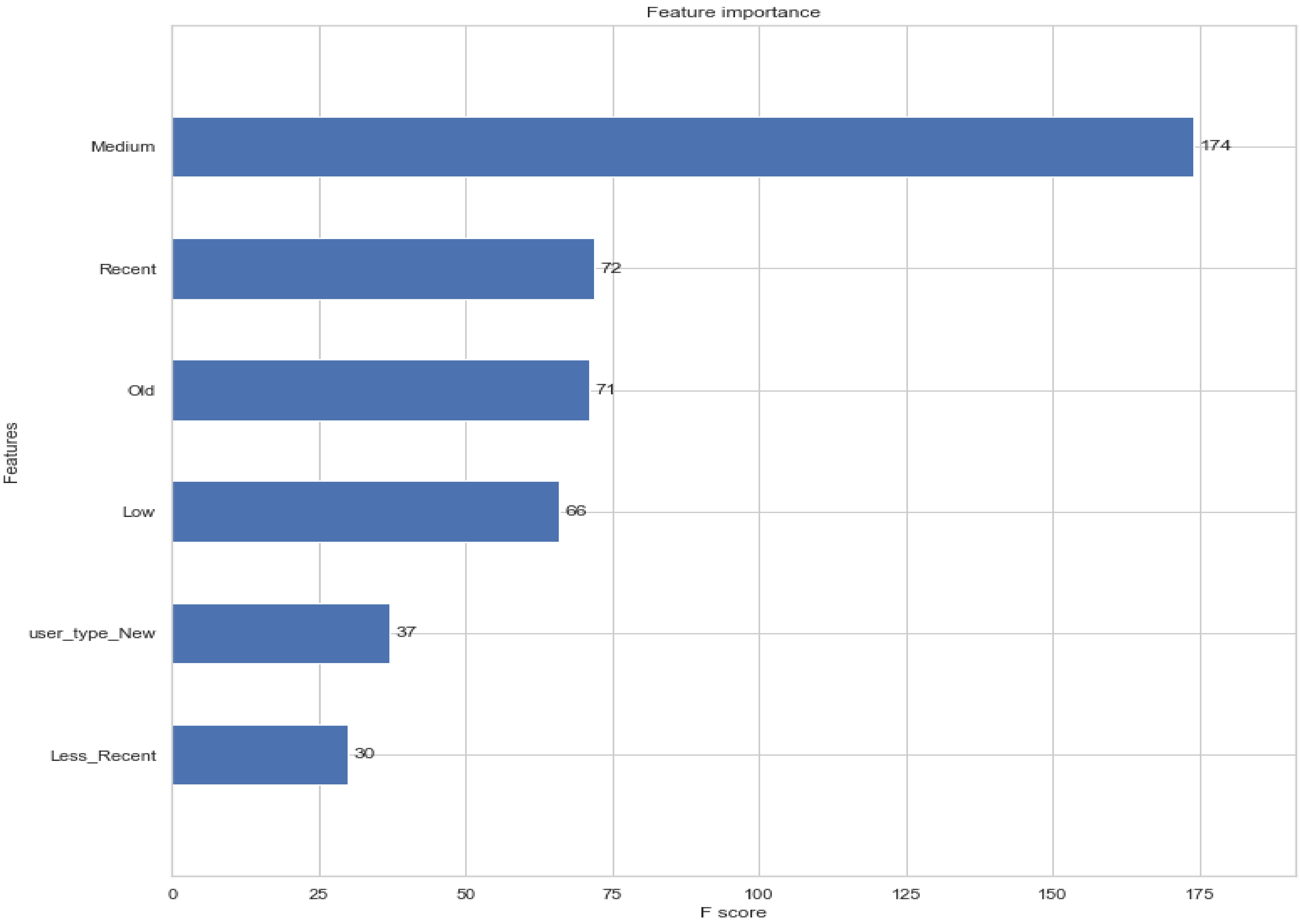
- The target variable revenue is not normally distributed to use Linear regression algorithm.
- Created a base Decision Tree model and as the number of trees increased, the average prediction error decreased.

XGBoost Performance



- Used an XGBoost ensemble to create the best model.
- Used GridSearchCV to identify the best parameters.
- Log Loss Curve indicates no overfitting.

```
mse: 2278.3776892635356
Actual_Revenue: 2149498.27
Predicted_Revenue: 2156111.0
```


Feature importance
Medium
Recent
Old
Low
user_type_New
Less_Recent
174
72
71
66
37
30
Features
0
25
50
75
100
125
150
175
F score

Summary

- This ensemble model can now be used to predict future revenue.

- As the number of existing customers were less, I haven't included any related feature to cluster audience segments.

Areas of Improvement:

- The geographical locations of customers and sellers can determine why customers in few areas are purchasing more compared to the others.

- Creating new segments with respect to customer behavior, channels can improve model’s performance.