# Analyzing eCommerce Customer Dynamics: Segmentation and Churn Analysis in Online Retail

William Hargis
University of Colorado - Boulder
william.hargis@colorado.edu

## ABSTRACT

In this paper, we describe an applied project in customer segmentation and churn analysis for retail electronic commerce. In the domain of eCommerce, customer retention remains an important challenge to competing retail businesses. This paper presents an applied project utilizing data mining techniques to perform customer segmentation and churn analysis for an online retail dataset. Using UCI Machine Learning Repository's Online Retail II dataset, this project will first employ RFM analysis for categorizing customer behavior. Then, various clustering algorithms will be applied to extract patterns of consumer behavior and enable more nuanced customer segmentation. The insights derived from this segmentation will then inform the development of machine learning models designed to predict customer churn. Preliminary evaluations will focus on the stability and efficacy of the segmentation, as well as the accuracy and precision of the churn prediction models. The primary goal is to utilize this dataset to derive actionable insights which could be used to enhance customer retention strategies.

## CCS Concepts

**• Applied computing→Electronic commerce • Computing methodologies→Machine learning.**

## Keywords

Electronic commerce, eCommerce, Data mining, Machine learning, Unsupervised learning, RFM (Recency, Frequency, Monetary) analysis, Clustering, Customer segmentation, Churn analysis.

## 1. INTRODUCTION

Electronic commerce comprises a dominant retail segment in the United States and has grown in the post-COVID period to over $1 trillion USD in annual sales as of 2022 [1]. This project seeks to use data mining methodologies to extract customer segmentation information to predict customer churn in a large electronic commerce dataset. Churn, the loss of an existing customer from a business' customer base, is of great interest to electronic commerce applications as in retail it is generally seen as true that new customer acquisition is more costly than retention of existing customers. Several methods for segmenting customers ranging from qualitative segmentation to clustering models are presented and compared. The results of our customer segmentation analysis are then employed to build several machine learning models for predicting customer churn and these resulting churn models are compared. Consideration is given to the scale of the dataset and identifying segments by consumer behavior patterns.

## 2. RELATED WORK

Data mining techniques for customer segmentation and retention management are a topic of common interest to the fields of data mining and business intelligence. There exists case studies employing RFM-analysis for customer segmentation [4] and of churn prediction using supervised learning techniques [5, 6].

Additionally, textbooks intended as methodological reference have been published with specific application to data mining techniques for customer retention management [3]. Tsiptsis and Chorianopoulos' methodological reference includes detailed sections on clustering using techniques such as KMeans.

## 3. DATASET

In this project we utilized the Online Retail II dataset provided by the UC Irvine Machine Learning Repository.[2] This dataset contains all transactions for an anonymized United Kingdom-based online-only retailer for the period of 01/12/2009 to 09/12/2011 which covers approximately 1 million transactions. This retailer primarily sells unique non-seasonal gift-ware with its primary customer base consisting of wholesalers, but does include direct-to-consumer sales. This dataset comprises more than 1 million rows of transaction data of over 5900 unique customers and therefore constitutes a fairly large dataset for our analysis.

## 4. PROPOSED WORK

In order to identify customer segments for targeted customer retention strategies, we will explore the dataset and perform a Recency, Frequency, Monetary (RFM) analysis – evaluate our RFM scores for qualitative labeling – and utilize the resulting the data for the preparation of clustering models on our dataset. Qualitative and clustering model derived labels will then be compared for their qualitative interest in targeting customer segments. Following this predictive classification models for customer churn will be prepared utilizing the RFM analysis as well as qualitative and clustering model derived labels as features. The predictive models will be evaluated based on their ability to correctly predict the churn of customers and their predicted probabilities for likelihood to churn. Furthermore, the usefulness of the engineered features to the predictive models will be qualitatively evaluated.

### 4.1 Customer Segmentation

#### 4.1.1 Exploratory Analysis

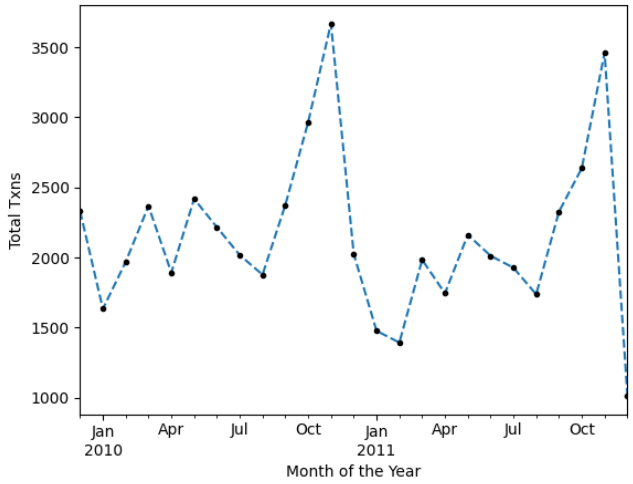
An exploratory analysis was conducted in which it was found that several hundred thousand rows of the dataset contain NaN entries for Customer ID or Description, these were dropped from the

dataset reducing the overall size to approx. 825,000 transactions and 5800 unique customers.

The dataset includes canceled orders alongside fulfilled orders, which must be taken into account when grouping transactions. Customers whose total spending was a negative balance due to the presence of only canceled order histories were dropped. The cleaned dataset showed the number of invoices was reduced by -16.3% and the number of transactions were reduced by -22.8%/

The dataset overall was evaluated with the following characteristics observed:

Transactions follow a seasonal pattern surrounding the end-of-year.



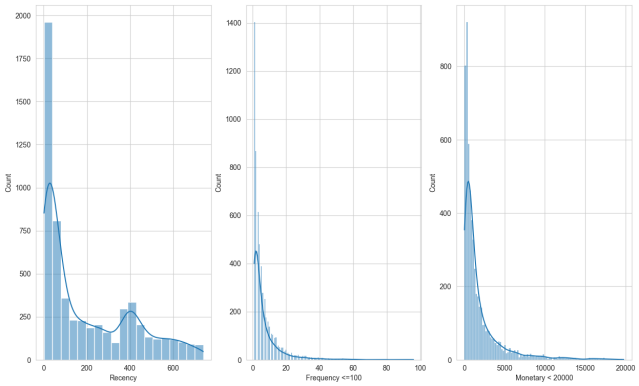

Transactions and total spend by country shows the majority of retail activity taking place in the UK, Germany, Ireland, Netherlands, and France. .

### 4.1.2 **RFM Analysis:**

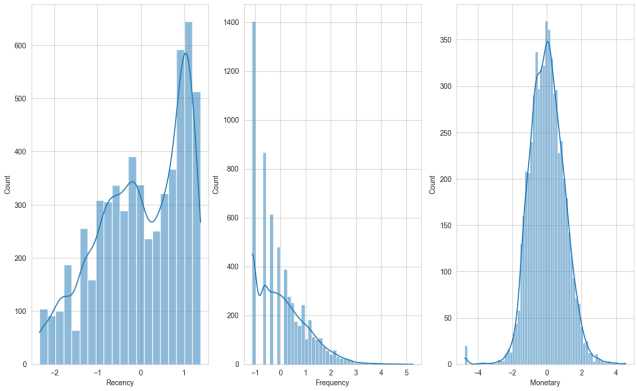
We performed an RFM (Recency, Frequency, Monetary) analysis as a preliminary step to categorize consumer behavior. This method has proven to be both simple and effective in other domains, and we observe similar results for eCommerce data.

We defined our Recency metric as the number of days since a customer's last transaction, our Frequency metric as a customer's number of unique transactions, and our Monetary metric as a customer's cumulative total spend. These metrics were then log-transformed and scaled before separating into quartile labels under R, F, and M headings. These quartile labels were used to form qualitatively segmented clusters of customer behavior.

Recency, Frequency, Monetary Distributions




Transformed Recency, Frequency, Monetary Distributions




Our RFM quartile labels present such that low R values are more recent, high F values are more frequent, and high M values represent greater total spend. Based on these we assigned the following qualitative RFM groupings:

**Proposed Qualitative Customer of Interest Groupings:**

| R | F | M | Quality |
|---|---|---|---|
| 1 | 1 | X | New Customer |
| 1 | 4 | X | High Activity |
| X | 4 | 4 | High Loyalty, High Spend |
| 1 | 4 | 4 | Prime Customer |
| 3 | 4 | 4 | Moderate Recency, High Frequency, High Spend |
| 4 | X | 4 | High Value, Inactive |
| 4 | 4 | X | High Loyalty, Inactive |
| 4 | (4, 3) | (4, 3) | High Loyalty, High Spend, Inactive |
| > 2 | <= 2 | <= 2 | Potential Churn |

These qualitative groupings were further segmented by whether the grouping represented a high loyalty or at-risk of churn interest:

**Loyal Customers of Interest**:

- Prime Customer
- High Activity
- High Loyalty, High Spend

**At-Risk Customers of Interest**:

- Moderate Recency, High Frequency, High Spend
- High Value, Inactive
- High Loyalty, Inactive
- High Loyalty, High Spend, Inactive
- Potential Churn

Quality Distribution

```
Quality
Other                                0.418815
Potential Churn                      0.314666
High Activity                        0.126174
High Loyalty, High Spend             0.082295
New Customer                         0.037391
High Loyalty, High Spend, Inactive   0.010927
High Value, Inactive                 0.006829
High Loyalty, Inactive               0.002903
```

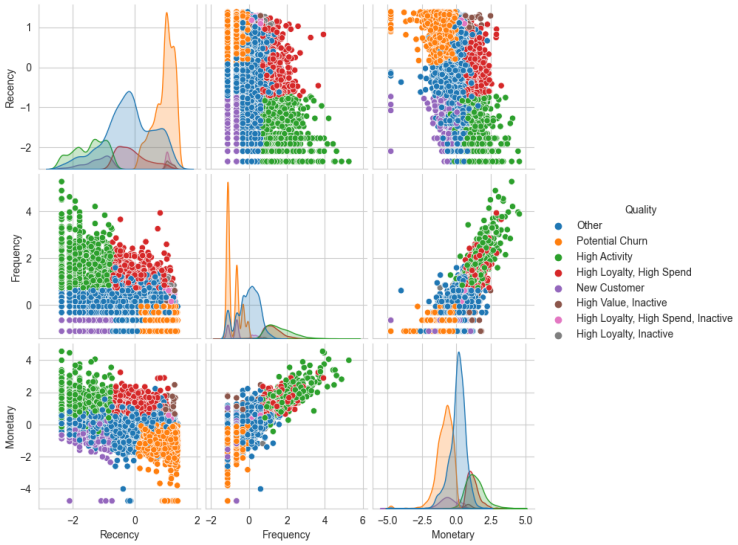
Our At-Risk segment comprised of Potential Churn categories:

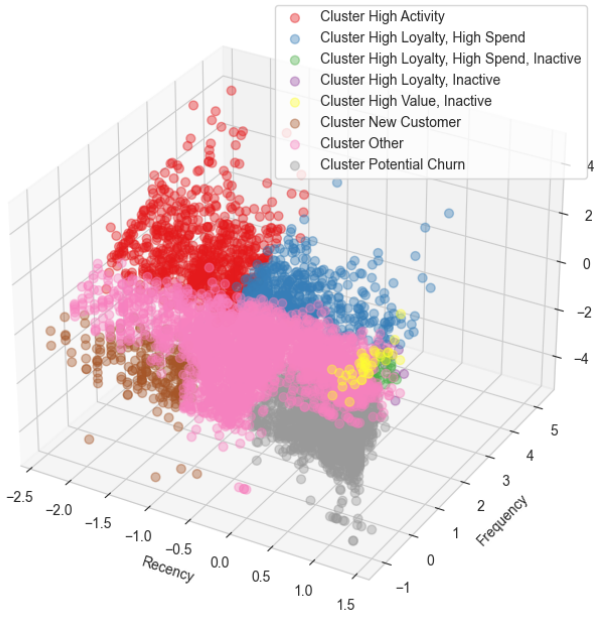
```
At_Risk
False     0.664675
True      0.335325
```

Our Loyalty Interest segment:

```
Loyal_Interest
False     0.791532
True      0.208468
```

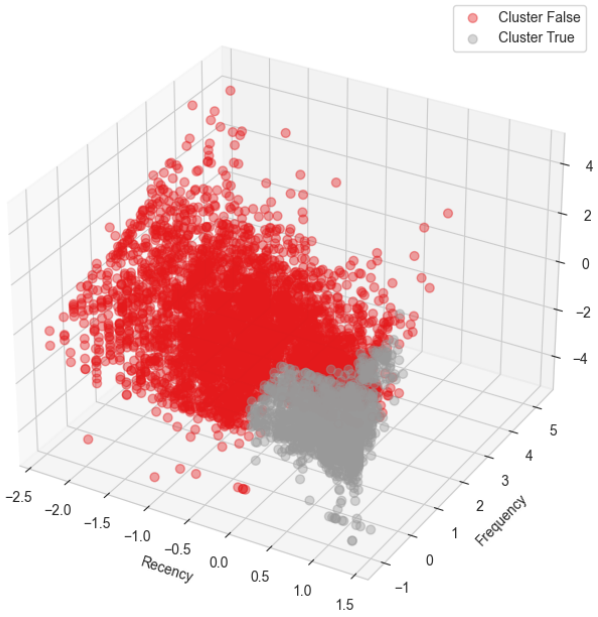
We visualized the RFM qualitative groupings on the three Recency, Frequency, and Monetary axes:




We observed distinct segmentation of the groupings across all three axes and believe that we have identified actionable groups for targeted customer retention strategies.

Of these groups the At-Risk segment forms a distinctive cluster of points at the vertex of high Recency and low Frequency:




From this we concluded that qualitatively grouping by RFM scores is producing meaningful segmentation and provides the opportunity to target the At-Risk segment we believe is likely to churn. This should also form an informed feature set for our later churn modeling and analysis.

### 4.1.3 **Clustering Algorithms:**

Building upon the RFM analysis, we employed a small number of distinct clustering algorithms to group similar customers based on their RFM characteristics. Algorithms considered include KMeans, DBSCAN, and Birch clustering. KMeans segments the dataset through minimizing inertia to form clusters of equal variance;

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(|x_i - \mu_j|^2)$$

DBSCAN segments the dataset by comparing point density to identify clustered regions as areas of high density surrounded by areas of low density;

DBSCAN selects an arbitrary seed point and expands clusters by finding all density-reachable points from the seed point based on an Eps-neighborhood of that point. The Eps-neighborhood is defined as the points within a given radius $\epsilon$ of the seed point.

$$\text{Nieghborhood}_\epsilon(p) - \{q \in D | \text{dist(p, q)} \leq \epsilon\}$$

Density reachability from point $p$ to point $q$ is defined as the existence of a chain of points $(p_1, \ldots, p_n)$ from $p_1 = q$ to the point $p_n = p$ where the intermediary points of the chain are directly density-reachable where

$$p_{n-1} \in \text{Neighborhood}_\epsilon(p_n)$$

and

$$\|\text{Neighborhood}_\epsilon\| \geq \text{Min. Points in Cluster}$$

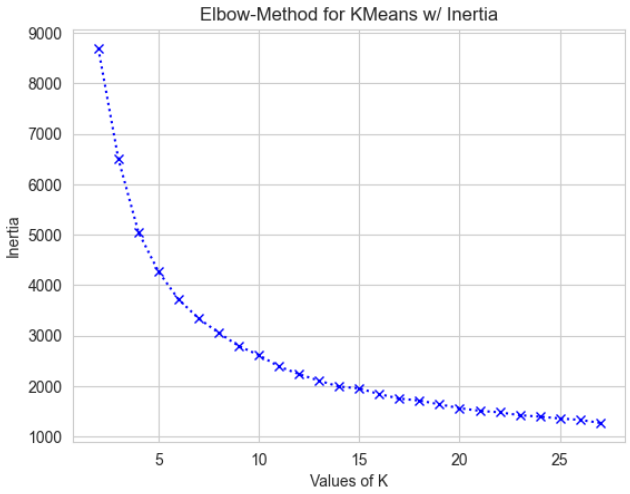
BIRCH clustering is a tree construction method where a Clustering Feature Tree is built based on merging point into branched nodes based on a radial distance. BIRCH constructs a hierarchical tree structure (CF Tree) to represent clusters. Each node in the tree summarizes a subset of data points using Clustering Features (CFs). The algorithm iteratively inserts points into the CF Tree to build from an empty tree to balanced, clustered branches. Clustering Features (CFs) of a node consist of the number of points $N$ in a node, the linear sum $\text{LSum}$ of the node's points, and the squared sum $\text{SSum}$ of the data points given as $CF = (N, \text{LSum}, \text{SSum})$.

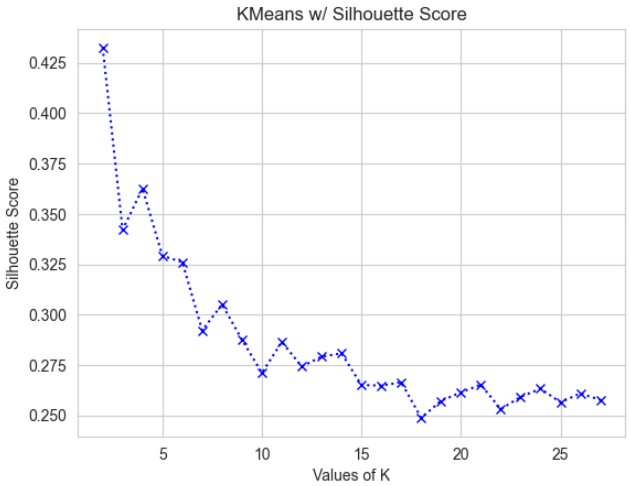
Points are inserted into nodes based on their minimal proximity to the centroid ($\text{Centroid} = \frac{\text{LSum}}{N}$) and nodes are branched once a predetermined point limit is reached.

The above algorithm descriptions are based on my interpretation of the scikit-learn user guide for clustering [7]

These clustering models remove the qualitative aspect of human selection in forming the clusters of interest.

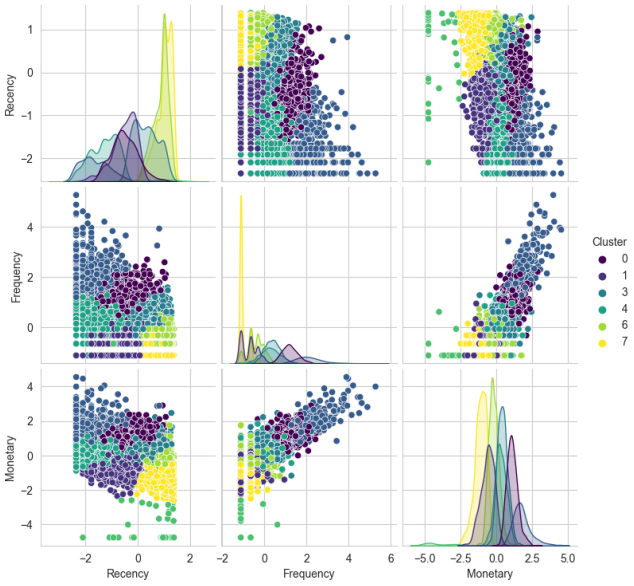
For KMeans and Birch clustering, the desired number of clusters must be specified to the model. The relevant number of clusters to be used was decided through the Elbow Method using Inertia in comparison with a graph of cluster silhouette scores. Silhouette scores measure intra-cluster similarity compared to inter-cluster similarity. Higher silhouette scores indicate distinction where there is cohesion, greater degrees of self-similarity of clusters, compared to separation, similarity between clusters.

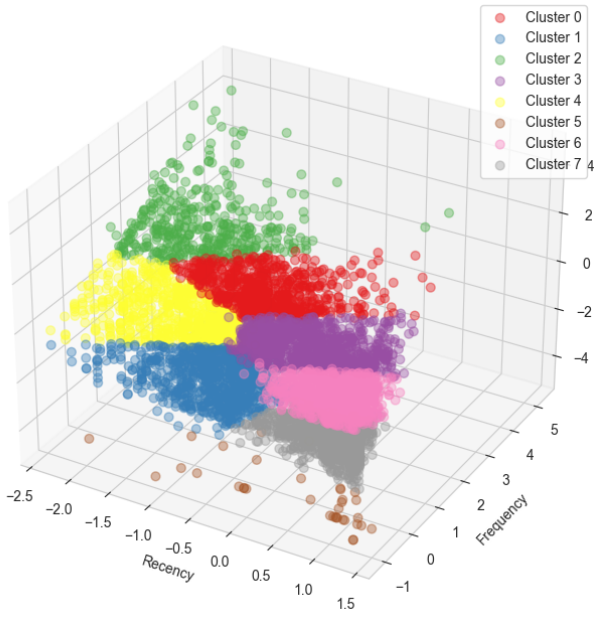



The elbow method paired with the silhouette scores lead us to choose a K value of 8. This value was utilized for both the KMeans and Birch clustering models.

**KMeans Clustering:** A KMeans clustering model was trained on the dataset and the predicted clusters were evaluated for cluster separation and distinction.





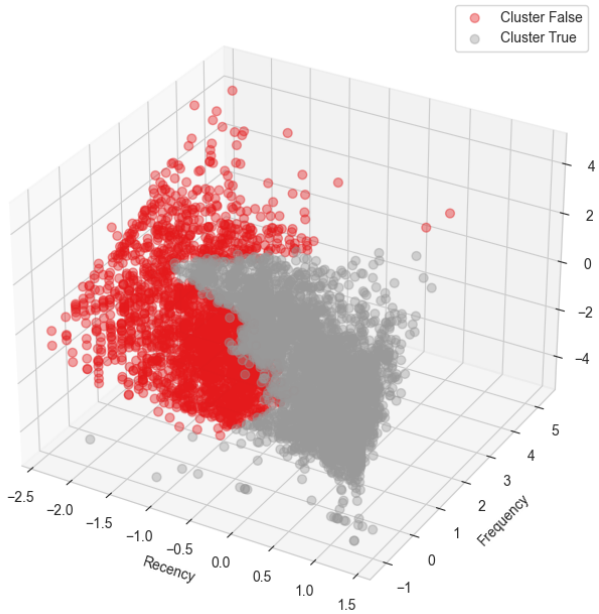

The KMeans cluster demonstrated distinct separation between clusters across all three Recency, Frequency, and Monetary axes.

These clusters were qualitatively evaluated to the following customer segments:

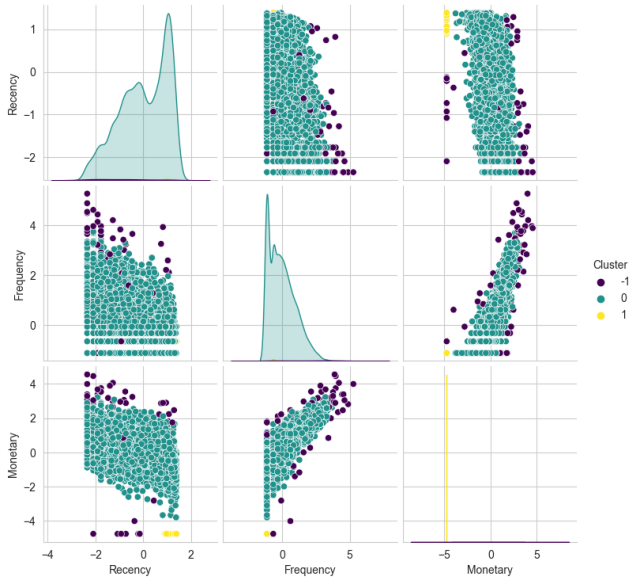
**Assumptions from plot:**

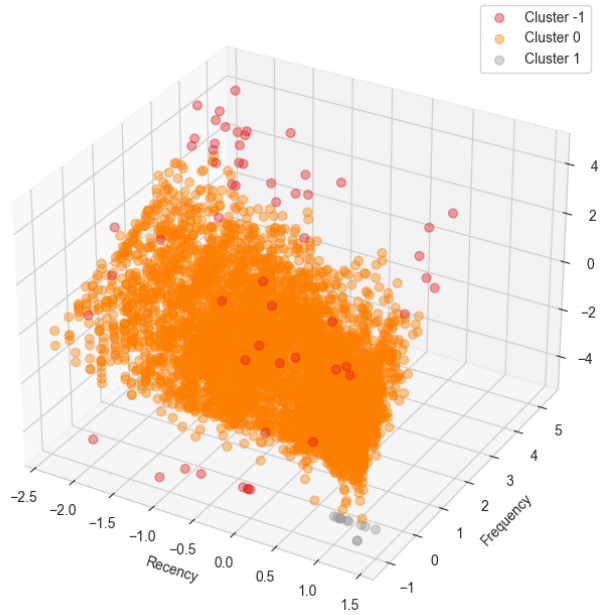
| Cluster | Cluster Description | Customer Description |
|---|---|---|
| 0 | Active, High Frequency, High Spend | At-Risk: Inactive Prime Customer |
| 1 | Active, Low Frequency, Low Spend | New Customer |
| 2 | Very Active, Highest Frequency, Highest Spend | Prime Customer |
| 3 | Moderately Inactive, Moderate in all categories | At-Risk: Inactive Normative Customer |
| 4 | Very Active, Moderate Frequency, Moderate Spend | Active Normative Customer |
| 5 | Moderate Inactivity, Low Frequency, Lowest Spend | At-Risk: Churned New Customers |
| 6 | Inactive, Low Frequency, Normal Spend | At-Risk: Likely-to-Churn Normative Customer |
| 7 | Inactive, Lowest Frequency, Low Spend | At-Risk: Churned Customers |

The At_Risk segmentation of the dataset appears over-zealous and may be grabbing too many of the high-spend customer segments without regard to recency:




**DBSCAN Clustering:** A DBSCAN clustering model was trained on the dataset and the predicted clusters were evaluated for cluster separation and distinction.

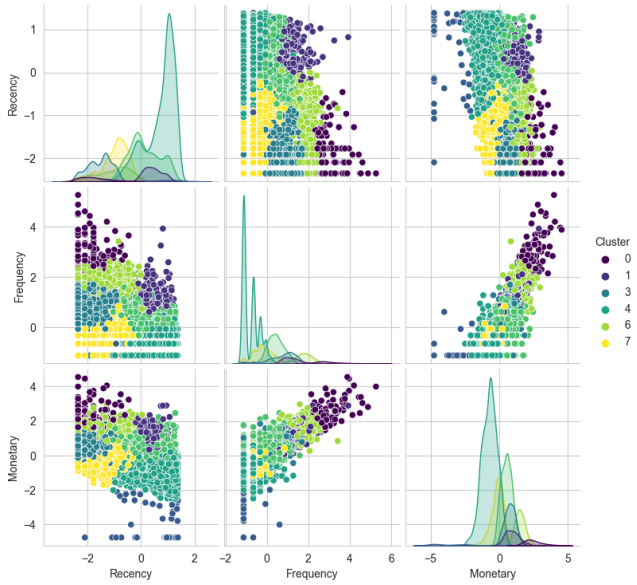


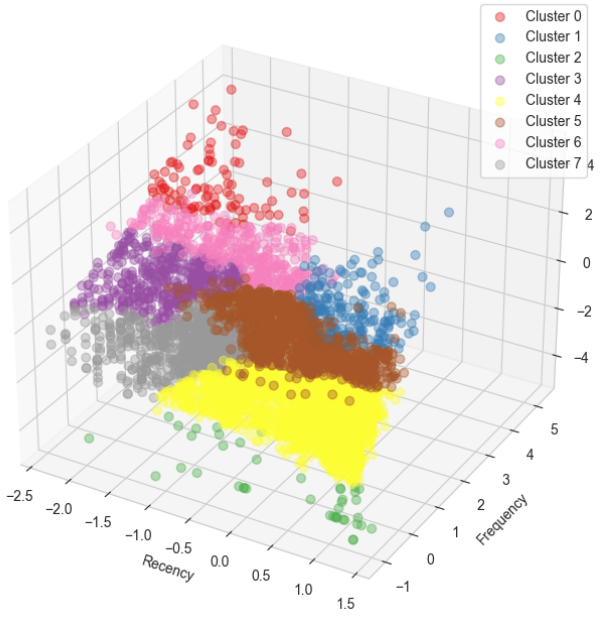




The DBSCAN cluster showed a lack of proper separation between clusters across the three axes. The DBSCAN model largely segmented the outliers from a rough plane in the 3D space spanned by the axes, where outliers were broadly determined using the Monetary axis. This follows from our understanding of the DBSCAN algorithm segmentation process where the high density plane has been identified alongside sparse clusters separated by low-density space. The resultant clusters are insufficient for our purposes and results did not appear to improve across a range of DBSCAN cluster parameters.

**BIRCH Clustering:**

A BIRCH clustering model was trained on the dataset and the predicted clusters were evaluated for cluster separation and distinction.




The Birch cluster demonstrated distinct segmentation across the three axes with similar outcomes to the KMeans clusters.

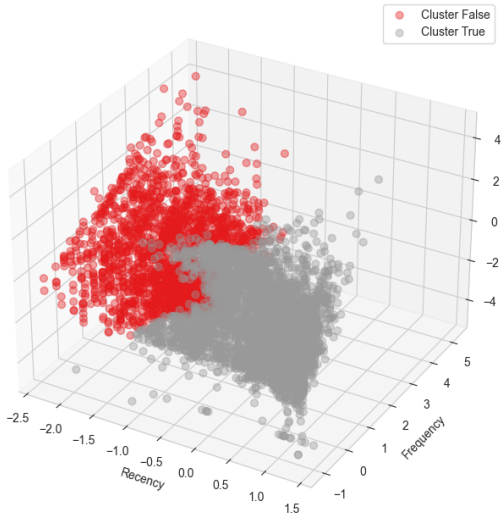
A notable difference between the Birch clustering and KMeans clustering being that the Birch model showed a greater sensitivity to the Recency axis with greater cluster separation across this axis.

These clusters were qualitatively evaluated to the following customer segments:

**Assumptions from plot:**

| Cluster | Cluster Description | Customer Description |
|---|---|---|
| 0 | Very Active, High Frequency, High Spend | Prime Customer |
| 1 | Inactive, High Frequency, High Spend | At-Risk: Inactive Prime Customer |
| 2 | Moderate Inactivity, Low Frequency, Lowest Spend | At-Risk: Inactive New Customers |
| 3 | Very Active, Moderate Frequency, Moderate Spend | Active Normative Customer |
| 4 | Inactive, Low Frequency, Low Spend | At-Risk: Likely to Churn Customer |
| 5 | Moderate Inactivity, Low Frequency, Moderate Spend | At-Risk: Inactive Normative Customer |
| 6 | Active, Moderate Frequency, High-Moderate Spend | Active Sub-Prime Customer |
| 7 | Active, Low Frequency, Low Spend | Active New Customers |

The At_Risk qualitative segmentation for the Birch clusters showed a decent separation based on recency, but like the KMeans clusters we may have qualitatively mapped too large of a segment into the At_Risk category:




In comparison to the KMeans clusters, the BIRCH clusters give qualitatively better targeted segmentation within the At-Risk segment based on customer spend and recency.

## 4.2 Churn Analysis

For the purpose of this case study, we have defined 'churned' customers as those who have not had any transactions within a 90 day period as this should represent a customer that has not interacted with the eCommerce platform within an approximate business quarter.

We applied a binary churn label to the prepared data from the RFM analysis and then prepared predictive models from multiple supervised learning algorithms.

### 4.2.1 Supervised Learning Algorithms:

**Log-Linear Classifier** or Logistic Regression for Classification is a linear model which seeks to minimize, in the binary case, the following cost function:

$$\min_{w} C \sum_{i=1}^{n} (-y_i \log(\hat{p}(X_i)) - (1 - y_i) \log(1 - \hat{p}(X_i))) + r(w)$$

where $r(w)$ is a regularization term adding a penalty to the model, we have left this regularization penalty as None.

**Stochastic Gradient Descent** is a means for fitting linear classifiers and regressors using the stochastic gradient descent process. The SGD process seeks to iteratively minimize a loss function $L(y_i, \hat{y}_i)$ subject to a regularization penalty $R(\theta)$. The gradient of the loss function is calculated with respect to the given model parameters at a selected data point. The parameters $\theta$ are then updated in the opposite direction of the gradient subject to a learning rate $\alpha$. For a cost function $J(\theta)$ and model prediction function $f(x_i, \theta)$, the gradient process looks like:

$$J(\theta) = -L(y_i, \hat{y_i}) \cdot f(x_i, \theta) + R(\theta)$$

where the model parameter update is given by

$$\theta = \theta - \alpha J(\theta)$$

This iterative process repeats up to the point of convergence or the iteration step limit.

**K-Nearest Neighbors** is a model that predicts the class label $C$ for a new data point based on the majority class among the new points K-nearest neighbors, this is analogous to a majority vote within a neighborhood of points.

This takes the form

$$C = \text{argmax}_{ci} \sum_{i=1}^{k} I(y_i = c_i)$$

**Decision Tree** is a model that recursively partitions the dataset based on feature values for each class. The tree formed through this splitting process is then traversed for each new point and a label is assigned based on the evaluated feature conditions.

**Random Forest** is an ensemble model in which several decision trees are constructed as weak classifiers and a majority vote of the individual decision tree predictions is used to find a consensus prediction. This consensus prediction is a stronger classifier than the individual decision trees.

**Support Vector Classification** is a model that aims to find the optimal hyperplane that separates the different classes within the dataset's feature space. The hyperplane takes the form

$$w \cdot x + b = 0$$

where $w$ is the weight vector defining the directionality of the hyperplane, $x$ is the feature vector of a data point, and $b$ is the bias or intercept term that shifts the hyperplane away from the origin. The optimal hyperplane is found by minimizing the margin $M$ between classes, where the margin is given by

$$M = \frac{2}{|w|}$$

and subject to

$$y_i(w \cdot x + b) \geq 1 \forall i$$

**Gradient Boost Classification** is an ensemble method where an ensemble of weak learners, in this case decision trees, are built sequentially subject to an iterative boosting process. Residuals for each point are calculated $r_i = y_i - F_{k-1}(x_i)$ where each $F_{k-1}(x_i)$ is the prediction from the prior step's ensemble, a decision tree is fitted to the residuals and the ensemble prediction is updated in the form

$$F_k(x) = F_{k-1}(x) + \eta * h_k(x)$$

where $\eta$ is the learning rate and $h_k(x)$ is the current weak learner/tree's prediction. This iterative process repeats until a stopping criterion or iteration limit is reached. The final prediction takes the form of the $\eta$ weighted majority vote of the weak learners.

The above algorithm descriptions are based on my interpretation of the scikit-learn user guide for supervised learning. [9]

**CatBoost** is a method of Gradient Boosting that incorporates a method for encoding categorical features via ordered boosting which creates a statistically ordered mapping of the categorical features based on their relation to the target variable. [10]

**Light Gradient Boost** is a method of Gradient Boosting that incorporates a method for encoding categorical features via Gradient-based One-Side Sampling and builds its decision trees from a histogram of bins fitted to the continuous features. The tree construction is also dependent on a leaf-based construction minimizing loss at each level of the tree. [11]

**Model Comparison:** We prepared a set of models for each of the above supervised learning methods trained on each of the following datasets: RFM without qualitative labels, RFM with qualitative labels, RFM with KMeans cluster labels, and RFM with BIRCH cluster labels. Cluster numerical labels were mapped to qualitatively evaluated customer segments provided in earlier tables. Each model was fitted and tuned using KFold cross validation to arrive at the best possible model on the test set.

# 5. EVALUATION

## 5.1 Metrics

Our **clustering** task involves the labeling of an unsupervised dataset of points, a scenario in which the ground truth labels for our points are not known. For the purposes of cluster evaluation we utilized the following metrics;

**Silhouette Score** is a coefficient measuring separation made up of two scores:

$a$ - the average distance between a sample and the other points in its class.

$b$ - the average distance between a sample and the points in the next nearest cluster.

The Silhouette Score for a sample is then given as

$$s = \frac{b - a}{\max(a, b)}$$

A greater silhouette score implies a better defined cluster.

**Calinski-Harabasz Score** is a ratio of the sum of inter-cluster dispersion and intra-cluster dispersion for all clusters; dispersion is the squared sum of distances.

For a set $E$ of size $n_E$ segmented into $k$ clusters, the score is defined as

$$s = \frac{tr(B_k)}{tr(W_k} * \frac{n_E - k}{k - 1}$$

where $tr(B_k)$ is the trace of the inter-cluster dispersion matrix and $tr(W_k)$ is the trace of intra-cluster dispersion matrix.

$$W_k = \sum_{q=1}^{k} \sum_{x_q} (x - c_q)(x - c_q)^T$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T$$

where $C_q$ is the set of points in cluster $q$ m $c_q$ is its center, and $c_E$ is the center of the set $E$ and $n_q$ is the number of points within cluster $q$.

A greater Calinski-Harabasz score implies better defined clusters.

**Davies-Bouldin Index** is a measure of similarity between clusters comparing the distance between clusters with the size of the clusters.

The index is given as the average similarity between clusters $C_i$ for $i = 1 \ldots k$ and its most similar cluster $C_j$. Similarity is defined as $R_{ij}$ where

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

for $s_i$ the average distance between each point of $C_i$ and $C_i$'s centroid and $d_{ij}$ is the distance between the cluster centroids $i$ and $j$.

The index is then defined as

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{ij} R_{ij}$$

The score is lower-bounded by zero and lower scores imply a better partition of clusters.

The scikit-learn user guide for clustering performance evaluation [7] was used for the above metric definitions.

Our **predictive modeling** task consists of our defined churn labels where churn is defined as a customer without a transaction within a 90 day period (approximately one business quarter). This supervised learning task evaluated the prepared models using the following metrics;

**Accuracy** is a measure of the fraction of true predictions out of the total predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

**Balanced Accuracy** is a measure of the average of recall scores per class, intended to prevent inflated performance from imbalanced data. This is equivalent to a class-weighted average of the standard accuracy score.

For the binary case, such as in our churning prediction, we have:

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right)$$

**Average Precision Score** is a measure that summarizes the **Area Under Curve - Precision Recall (AUC-PR)** as a weighted average of precision at each threshold. The increase in recall from each previous threshold is used as a weight.

$$AP = \sum_{n} (R_n - R_{n-a}) P_n$$

Here, $P_n$ and $R_n$ are the precision and recall scores at the given threshold $n$.

**Matthew's Correlation Coefficient (MCC)** is a measure of the quality of binary classifications. It takes the form of a correlation coefficient value between -1 and 1, where 0 is a random case, -1 is an inverse prediction, and 1 is a true prediction.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

**Cohen's Kappa** is a measure of inter-rater reliability for categorical ratings, it measures the degree of agreement between two raters classifying items into mutually exclusive categories.

For the binary case:

$$\kappa = \frac{2 * (TP * TN - FN * FP)}{(TP + FP) * (FP + TN) * (TP + FN) * (FN + TN)}$$

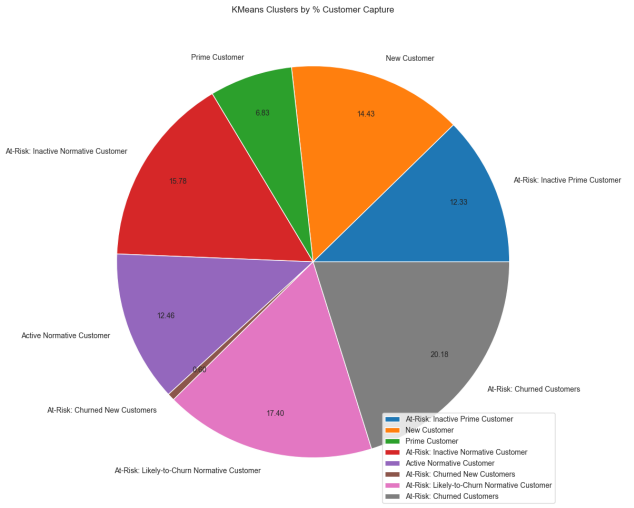
**Log Loss** is a measure of the difference between predicted probability estimates for a model and actual value. For the binary case where $y \in \{0, 1\}$ and $p = \Pr(y = 1)$ the negative per sample log-likelihood of the classifier compared to the true label:

$$L_{\log}(y, p) = -\log Pr(y|p) = -\left(y \log(p) + (1 - y) \log(1 - p)\right)$$

The scikit-learn user guide for model evaluation [8] was used for the above metrics.

## 5.2 Customer Segmentation

The customer segmentations derived from the qualitative labeling, KMeans clustering labeling, and BIRCH clustering labeling were compared. In terms of customer capture, the qualitative RFM labeling was found to hold the majority of its customers in the 'Potential Churn'(31.5%) and 'Other'/'Normative Customer' categories (41.9%) followed by the 'High Activity' category (12.6%) whereas these categories captured 3.5%, 19.7%, and 56.4% of the monetary activity. For KMeans clusters, the customer capture was fairly evenly distributed.




In terms of monetary capture, our KMeans 'Prime Customer' accounted for 51% with its equivalent at-risk/inactive category at 22.4%. For BIRCH clusters, the customer capture placed plurality of customers in the 'Likely to Churn' label (44.3%) with next largest label being the 'Inactive Normative Customer' (19.8%) and 'New Customer' (13.5%), the 'Prime Customer' and 'Sub-Prime Customer' captured 1.7% and 6.9%, respectively. In terms of monetary capture these labels captured 6.6%, 16.3%, 4%, 33.5%, and 22.1%.

The BIRCH labels provide the greatest segmentation based on the monetary axis giving greater targeting based on spend, KMeans identifies a more even customer capture and a large monetary capture of 'Inactive Prime', qualitative RFM identifies negligible monetary activity in its At-Risk segment, but identifies a large customer segment with extremely low monetary activity in the 'Potential_Churn'.

The RFM Qualitative labels place over 1,000 customers into the Not At-Risk segment that end up being churned with ~825 being normative customer and ~150 being High Loyalty High Spend customer labels. This presents a large cost in failing to identify these customers for targeted retention. In comparison the KMeans clusters only have 47 customers listed as Not At-Risk that end up being churned, 45 of which are New Customer labels and 2 are Prime Customers - the Prime Customer miss is likely due to lack of Recency sensitivity, but presents as a moderate cost. The BIRCH clusters do not have any of its Not At-Risk segment customers end up being churned, showing no cost in the labeling –likely due to the identified Recency sensitivity.

The inverse case of identifying customers to be At-Risk when they do not end up churning does not present itself as a costly error, so while the qualitative RFM labels have no false At-Risk labeling it is not a beneficial characteristic of the segmentation.

The clusters were also compared for distinction/separation of clusters:

| | RFM_Qual | KMeans | BIRCH |
|---|---|---|---|
| silhouette | 0.122222 | 0.305090 | 0.272082 |
| ch_score | 1689.169788 | 3968.247834 | 2631.747786 |
| db_score | 1.125971 | 0.938376 | 1.120567 |

From this, the most distinct clusters were found in the KMeans segmentation, followed by the BIRCH and RFM Qualitatively labeled clusters. The difference between the KMeans and BIRCH scores were not that great outside of the Calinski-Harabasz Score.

We determined from these evaluations that the BIRCH clustering model was the most performant for the purposes of identifying segments for targeted customer retention strategies as it had no false negatives and similar rates of false positives to the KMeans cluster. The False negatives for the Qualitative RFM were surprisingly high and present as a large business cost.

## 5.3 Customer Churn Analysis

All of the selected supervised learning models produced similarly performant models with only marginal improvements to models from the clustered features being included. Even when compared to the base model case of a Logistic Regression classifier trained on the RFM scores without qualitative labeling there is little to no observable improvement between models and engineered feature sets.

Raw RFM

| | Model | Accuracy | AUC-PR | MCC | Balanced Accuracy | Cohen's Kappa | Log Loss |
|---|---|---|---|---|---|---|---|
| 0 | LR | 0.991468 | 0.996502 | 0.983076 | 0.991424 | 0.982933 | 0.037147 |
| 1 | SGD | 0.991468 | 0.994398 | 0.983076 | 0.991424 | 0.982933 | 0.307540 |
| 2 | KNN | 0.991468 | 0.994311 | 0.983076 | 0.991424 | 0.982933 | 0.200438 |
| 3 | CART | 0.982935 | 0.974521 | 0.965869 | 0.982935 | 0.965869 | 0.615079 |
| 4 | RF | 0.990614 | 0.994822 | 0.981342 | 0.990575 | 0.981227 | 0.181115 |
| 5 | SVC | 0.991468 | 0.997770 | 0.983076 | 0.991424 | 0.982933 | 0.041815 |
| 6 | GBoost | 0.990614 | 0.997282 | 0.981342 | 0.990575 | 0.981227 | 0.039479 |
| 7 | CatBoost | 0.991468 | 0.997274 | 0.983076 | 0.991424 | 0.982933 | 0.038616 |
| 8 | LightGBM | 0.990614 | 0.997503 | 0.981342 | 0.990575 | 0.981227 | 0.050216 |

RFM w/ Qualitative Labeling

| | Model | Accuracy | AUC-PR | MCC | Balanced Accuracy | Cohen's Kappa | Log Loss |
|---|---|---|---|---|---|---|---|
| 0 | LR | 0.991468 | 0.996123 | 0.983076 | 0.991424 | 0.982933 | 0.038891 |
| 1 | SGD | 0.991468 | 0.991380 | 0.983076 | 0.991424 | 0.982933 | 0.307540 |
| 2 | KNN | 0.991468 | 0.993601 | 0.983076 | 0.991424 | 0.982933 | 0.229984 |
| 3 | CART | 0.982935 | 0.974521 | 0.965869 | 0.982935 | 0.965869 | 0.615079 |
| 4 | RF | 0.989761 | 0.993480 | 0.979612 | 0.989726 | 0.979520 | 0.234116 |
| 5 | SVC | 0.991468 | 0.997849 | 0.983076 | 0.991424 | 0.982933 | 0.041826 |
| 6 | GBoost | 0.990614 | 0.997329 | 0.981342 | 0.990575 | 0.981227 | 0.039248 |
| 7 | CatBoost | 0.991468 | 0.997363 | 0.983076 | 0.991424 | 0.982933 | 0.038392 |
| 8 | LightGBM | 0.991468 | 0.997573 | 0.983076 | 0.991424 | 0.982933 | 0.049981 |

RFM w/ KMeans Labeling

| | Model | Accuracy | AUC-PR | MCC | Balanced Accuracy | Cohen's Kappa | Log Loss |
|---|---|---|---|---|---|---|---|
| 0 | LR | 0.991468 | 0.998902 | 0.983076 | 0.991424 | 0.982933 | 0.027925 |
| 1 | SGD | 0.991468 | 0.994828 | 0.983076 | 0.991424 | 0.982933 | 0.307540 |
| 2 | KNN | 0.991468 | 0.996124 | 0.983076 | 0.991424 | 0.982933 | 0.138510 |
| 3 | CART | 0.985495 | 0.977106 | 0.971002 | 0.985508 | 0.970990 | 0.522817 |
| 4 | RF | 0.989761 | 0.998425 | 0.979527 | 0.989752 | 0.979521 | 0.059187 |
| 5 | SVC | 0.991468 | 0.998167 | 0.983076 | 0.991424 | 0.982933 | 0.037390 |
| 6 | GBoost | 0.993174 | 0.998873 | 0.986399 | 0.993148 | 0.986347 | 0.027217 |
| 7 | CatBoost | 0.994027 | 0.998957 | 0.988124 | 0.993997 | 0.988054 | 0.026388 |
| 8 | LightGBM | 0.992321 | 0.998871 | 0.984676 | 0.992299 | 0.984641 | 0.032852 |

RFM w/ BIRCH Labeling

| | Model | Accuracy | AUC-PR | MCC | Balanced Accuracy | Cohen's Kappa | Log Loss |
|---|---|---|---|---|---|---|---|
| 0 | LR | 0.991468 | 0.997228 | 0.983076 | 0.991424 | 0.982933 | 0.035768 |
| 1 | SGD | 0.991468 | 0.991380 | 0.983076 | 0.991424 | 0.982933 | 0.308120 |
| 2 | KNN | 0.990614 | 0.994331 | 0.981342 | 0.990575 | 0.981227 | 0.201140 |
| 3 | CART | 0.986348 | 0.981195 | 0.972717 | 0.986330 | 0.972695 | 0.492064 |
| 4 | RF | 0.985495 | 0.995868 | 0.971001 | 0.985481 | 0.970988 | 0.151847 |
| 5 | SVC | 0.991468 | 0.997770 | 0.983076 | 0.991424 | 0.982933 | 0.042305 |
| 6 | GBoost | 0.991468 | 0.998122 | 0.983076 | 0.991424 | 0.982933 | 0.036573 |
| 7 | CatBoost | 0.991468 | 0.997881 | 0.983076 | 0.991424 | 0.982933 | 0.036644 |
| 8 | LightGBM | 0.988908 | 0.997963 | 0.977884 | 0.988877 | 0.977814 | 0.050998 |

This may be due to the limited size of the RFM dataset, though the test sample does contain well over 1000 customers. From the Log Loss scoring, however, the best performing model for determining likelihood of a customer to churn is the Logistic Regression Classifier with the KMeans Labels set, though it is very similar in performance to several other model and labeling combinations. From the Confusion Matrices (see SI) the CatBoost model with KMeans labels produced the fewest false negatives and false positives, though this difference is in the small single digits from other model and label combinations.

# 6. DISCUSSION

Our case study was able to evaluate several clustering strategies for use in targeted customer retention and many machine learning methods for use in predicting customer churn. In the process of running our analyses it was clear that some of the chosen clustering methods were inappropriate for the dataset's three dimensional feature space as the DBSCAN model identified a singular high density space and outlying low density clusters which proved insufficient for our intended use-case. The clustering methods were evaluated in light of the high costs of false negatives for risk identification as these show customers who would not be reached by targeted retention strategies. In evaluating our clustering methods it was determined that the BIRCH clustering method produced the best outcome for the intended business intelligence use case as it produced no false negatives. The KMeans clustering's false negatives were especially costly in that they included a portion of identified highest value/prime customers which have the greatest monetary loss to the retail enterprise if they churn.

The churn prediction models had a surprising result in that all models were able to performantly predict the churn of customers based on the RFM analysis dataset. This may be due to the simplicity of the primary features, the existence of the quartile labels in all datasets or the limitations of the size of the dataset.

Future work may include evaluating additional clustering methods on the RFM analysis dataset or the preparation of transactional level churn analyses based on rolling RFM measures and rolling churn. The transactional case may provide for better distinction between the supervised learning methods and would provide greater temporal sensitivity in identifying at-risk customers and their churn likelihood. The current case study is temporally limited by grouping by customer id to get overall customer behavior in terms of the Recency, Frequency, and Monetary axes we defined. Extending the methods shown to additional datasets may also be worthwhile, as larger or more diverse retail sets may provide greater challenges in modeling churn that weren't encountered by our current modeling strategies on the Online Retail II dataset. It may also be useful to extend the scope of the data mining and prediction strategy to predict a customer's lifetime value to the retailer based on their transactional or RFM data.

# 7. CONCLUSION

In our case study we were able to extract meaningful customer segmentations for targeted customer retention strategies with the BIRCH clustering method in particular resulting in a clustering model that labels our At-Risk customers without producing false negatives. Our case study showed great value in the RFM Analysis methodology and also demonstrated the ease with which this analysis method presents for churn prediction across a range of supervised learning methods. The supervised learning methods in our case study were able to achieve high precision, recall, and accuracy in all modeling cases.

# 8. REFERENCES


[1] U.S. Census Bureau. 2023. Quarterly Retail E-Commerce Sales. (August 17, 2023). Retrieved September 28, 2023 from https://www.census.gov/retail/ecommerce.html.

[2] Daqing Chen. 2019. Online Retail II. UCI Machine Learning Repository. Retrieved from https://doi.org/10.24432/C5CG6D.

[3] K. K. Tsiptsis and A. Chorianopoulos. 2011. Data Mining Techniques in CRM: Inside Customer Segmentation. Wiley, Germany.

[4] Chen, D. 2012. Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining. Journal of Database Marketing and Customer Strategy Management, 19(3), 197-208. ttps://doi.org/10.1057/dbm.2012.17.

[5] Hemlata Jain, Ajay Khunteta, and Sumit Srivastava. 2020. Churn Prediction in Telecommunication using Logistic Regression and Logit Boost. Procedia Computer Science 167 (2020), 101-112. https://doi.org/10.1016/j.procs.2020.03.187.

[6] Seema Baghla and Gaurav Gupta. 2022. Performance Evaluation of Various Classification Techniques for Customer Churn Prediction in E-commerce. Microprocessors and Microsystems 94 (2022), 104680. https://doi.org/10.1016/j.micpro.2022.104680.

[7] scikit-learn contributors. Scikit-Learn Documentation. Clustering. Retrieved from scikit-learn User Guide (URL: https://scikit-learn.org/stable/modules/clustering.html).

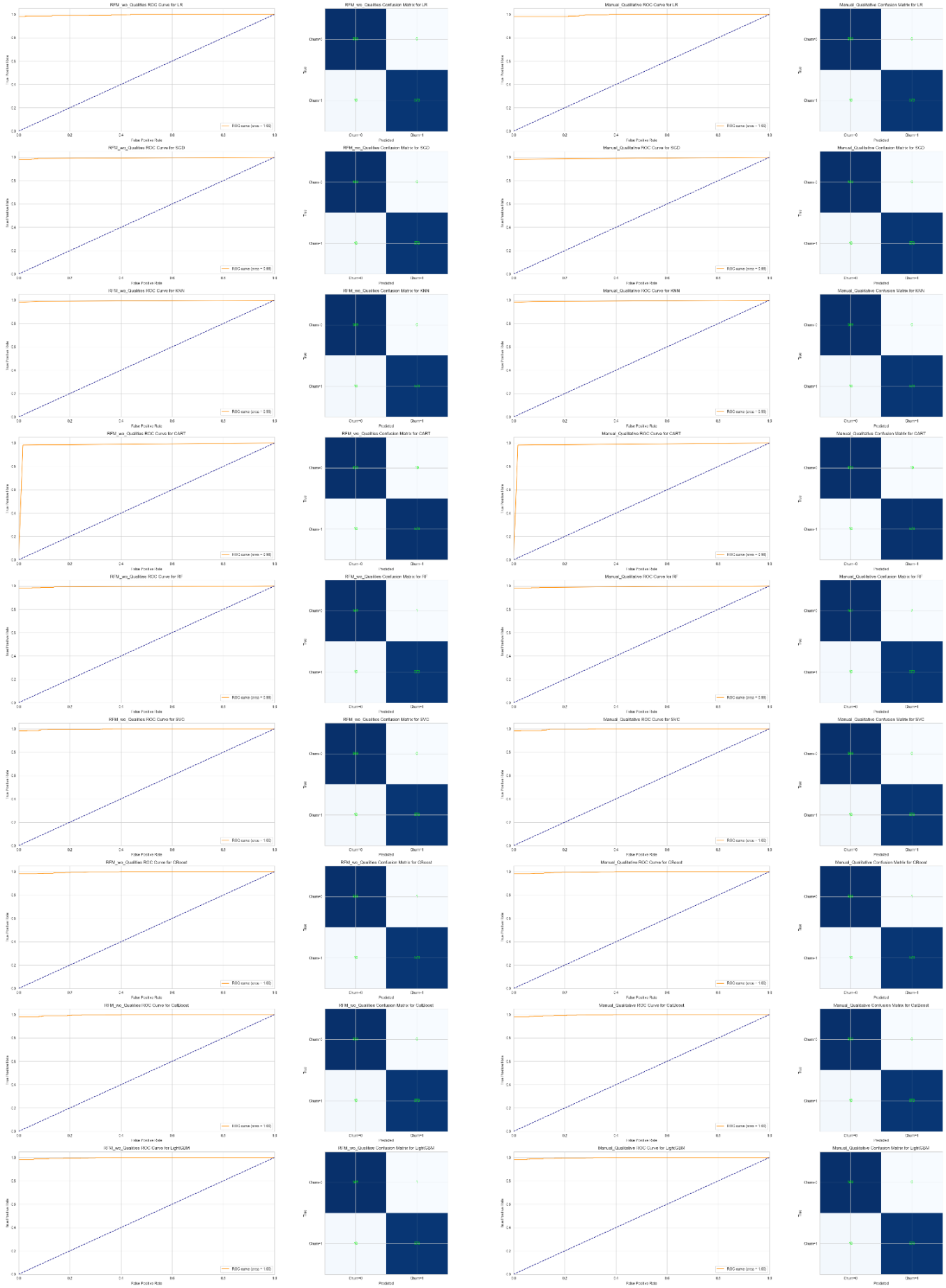
[8] scikit-learn contributors. Scikit-Learn Documentation. Metrics and scoring: quantifying the quality of predictions. Retrieved from scikit-learn User Guide (URL: https://scikit-learn.org/stable/modules/model_evaluation.html).

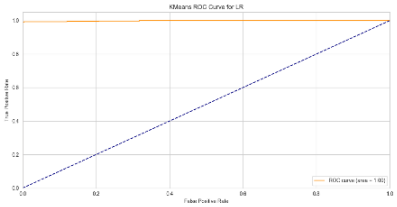
[9] scikit-learn contributors. Scikit-Learn Documentation. Supervised Learning. Retrieved from scikit-learn User Guide (URL:https://scikit-learn.org/stable/supervised_learning.html).

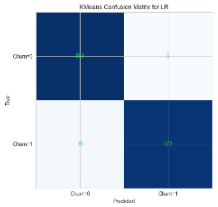
[10] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., & Gulin, A. 2018. CatBoost: Unbiased Boosting with Categorical Features. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS '18), 6639-6649. Retrieved from http://papers.neurips.cc/paper/7898-catboost-unbiased-boosting-with-categorical-features.pdf

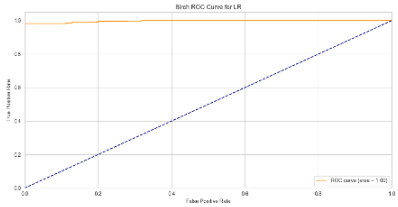
[11] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS '17), 3146-3154. Retrieved from https://proceedings.neurips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf

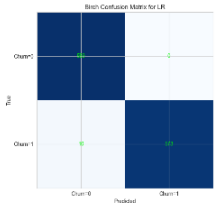
# 9. SUPPLEMENTARY INFORMATION

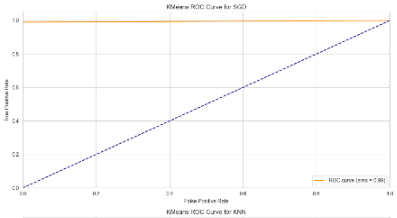


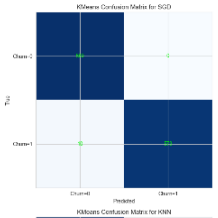

KMeans ROC Curve for LR
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)

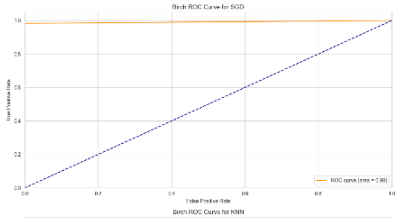

KMeans Confusion Matrix for LR
True
Predicted
Churn=0
Churn=1

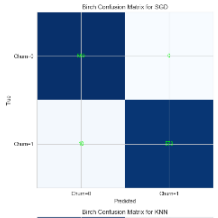

Birch ROC Curve for LR
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)

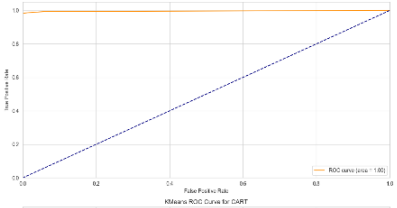

Birch Confusion Matrix for LR
True
Predicted
Churn=0
Churn=1

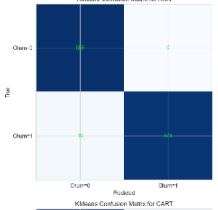

KMeans ROC Curve for SGD
True Positive Rate
False Positive Rate
ROC curve (area = 0.99)

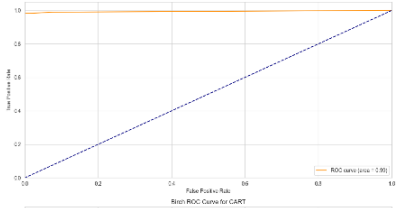

KMeans Confusion Matrix for SGD
True
Predicted
Churn=0
Churn=1

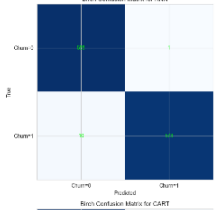

Birch ROC Curve for SGD
True Positive Rate
False Positive Rate
ROC curve (area = 0.99)

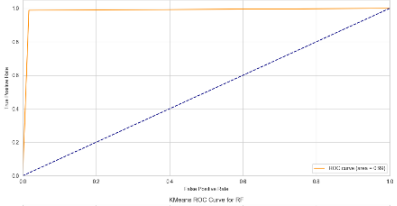

Birch Confusion Matrix for SGD
True
Predicted
Churn=0
Churn=1

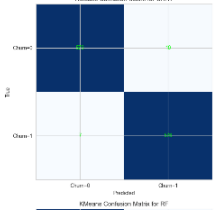

KMeans ROC Curve for KNN
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)

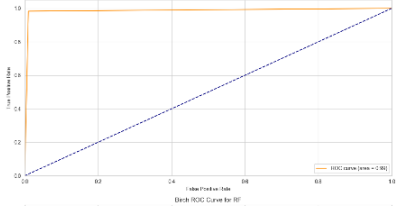

KMeans Confusion Matrix for KNN
True
Predicted
Churn=0
Churn=1

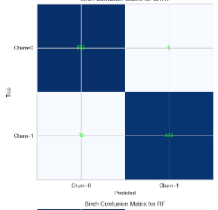

Birch ROC Curve for KNN
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)

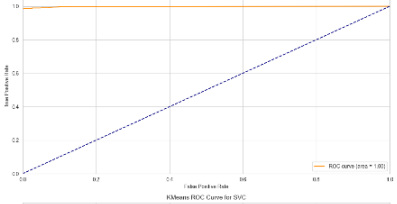

Birch Confusion Matrix for KNN
True
Predicted
Churn=0
Churn=1

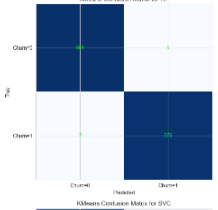

KMeans ROC Curve for CART
True Positive Rate
False Positive Rate
ROC curve (area = 0.99)

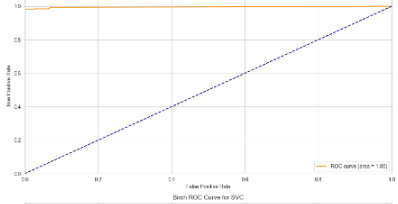

KMeans Confusion Matrix for CART
True
Predicted
Churn=0
Churn=1

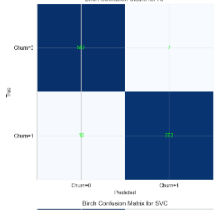

Birch ROC Curve for CART
True Positive Rate
False Positive Rate
ROC curve (area = 0.99)

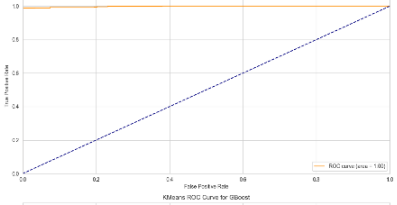

Birch Confusion Matrix for CART
True
Predicted
Churn=0
Churn=1

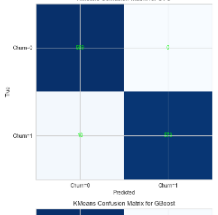

KMeans ROC Curve for RF
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)

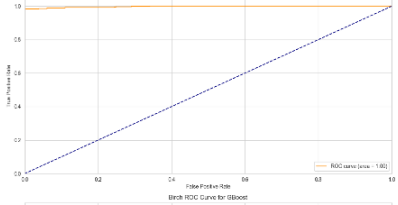

KMeans Confusion Matrix for RF
True
Predicted
Churn=0
Churn=1

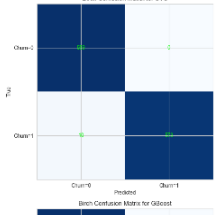

Birch ROC Curve for RF
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)

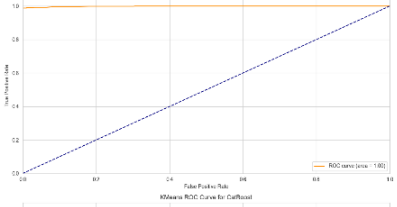

Birch Confusion Matrix for RF
True
Predicted
Churn=0
Churn=1

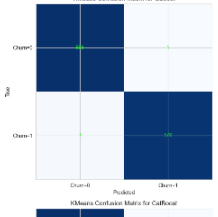

KMeans ROC Curve for SVC
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)

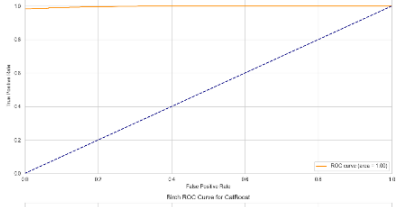

KMeans Confusion Matrix for SVC
True
Predicted
Churn=0
Churn=1

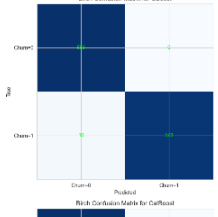

Birch ROC Curve for SVC
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)

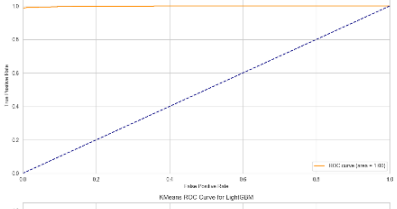

Birch Confusion Matrix for SVC
True
Predicted
Churn=0
Churn=1

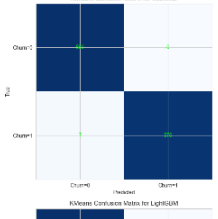

KMeans ROC Curve for GBoost
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)

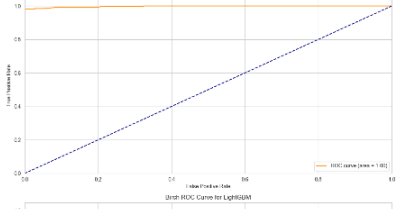

KMeans Confusion Matrix for GBoost
True
Predicted
Churn=0
Churn=1

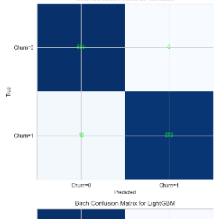

Birch ROC Curve for GBoost
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)

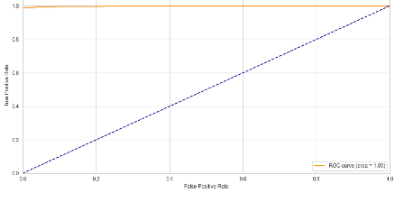

Birch Confusion Matrix for GBoost
True
Predicted
Churn=0
Churn=1

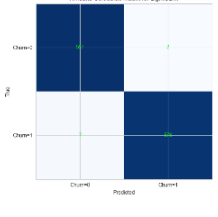

KMeans ROC Curve for CatBoost
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)

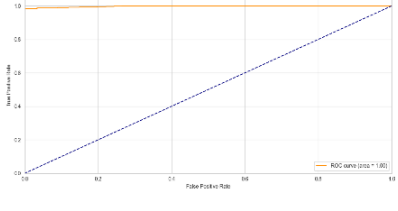

KMeans Confusion Matrix for CatBoost
True
Predicted
Churn=0
Churn=1

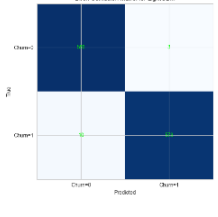

Birch ROC Curve for CatBoost
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)


Birch Confusion Matrix for CatBoost
True
Predicted
Churn=0
Churn=1


KMeans ROC Curve for LightGBM
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)


KMeans Confusion Matrix for LightGBM
True
Predicted
Churn=0
Churn=1


Birch ROC Curve for LightGBM
True Positive Rate
False Positive Rate
ROC curve (area = 1.00)


Birch Confusion Matrix for LightGBM
True
Predicted
Churn=0
Churn=1