Estimated Expected Customer Value using sklearn and PySpark

CIS731 Final Project Report

Kaitlyn Stick-Mueller

December 13, 2024

**Introduction**

Marketing is one of the most important components of any business. Marketing to the wrong customers can cost businesses large amounts of money and loss of valuable customers. Choosing to market to the right customers can help a business succeed.

The immediate assumption of determining the “right” customers would be to focus on who has spent the most money at the business. However, many marketing models suggest that there are more factors necessary to successfully determine customer value.

Customer Lifetime Value is an important metric to ensure businesses invest in the right customers. It allows businesses to focus on the customers who are loyal to the brand and likely to continue their relationship with the company (Saker et al., 2020). Some marketing models suggest focusing on a customer’s recency, frequency, and monetary value with the company (Murphy, 2024). Focusing on these three metrics (called RFM), allows businesses to focus on each customer’s purchasing patterns. The recency component focuses on how recent a customer has purchased from a company, frequency considers how many purchases the customer has made, and monetary value is composed of how much money the customer has spent at the company (Murphy, 2024). Understanding these RFM components of a customer can allow businesses to predict their valuable and invaluable customers. Throughout this project, using PySpark Machine Learning models, customer valuableness was predicted based on their RFM purchasing habits by clustering and then classifying customers.

**Background & Related Work**

This project was inspired by the Databricks Solution Accelerator “Analyze Customer Lifetime Value.” The Solution Accelerator provided a starting point for the project, with suggestions on cleaning, exploring, and preparing the dataset for modeling. It also provided guidance about the uses of the RFM metrics. Further understanding of the RFM metrics and their uses in business marketing was found in Casey Murphy’s 2024 article *What Is Recency, Frequency, Monetary Value (RFM) in Marketing?,* including its business relation and calculation*.* Insights about the importance of Customer Lifetime Value were also explored within Databricks’ article accompanying their Solution Accelerator, which discussed how Customer Lifetime Value is an important component of proper marketing (Saker, et al., 2020).

The dataset used in this project is titled “Online Retail,” and it was provided through the Databricks Solution Accelerator. The dataset comes from the UC Irvine Machine Learning Repository, and it is comprised of 491,909 rows of transaction data from one year at a UK based online store (Chen, 2015). Each row of the dataset includes a separate item from each invoice, and includes the item’s invoice number, stock code, description, quantity, invoice date, unit price, customer ID, and country of purchase. The data was collected from December 2010 to December 2011. Each line describes a unique item of an invoice. Some components of the data had to be further cleaned, as discussed in the Methodology section. The dataset is unlabeled, so there is no direct information provided about a customer’s value. Thus, the first task within the project is to cluster the customers based on their RFM components. Using their clusters as labels, classification models can then be used to predict customer value to the company.

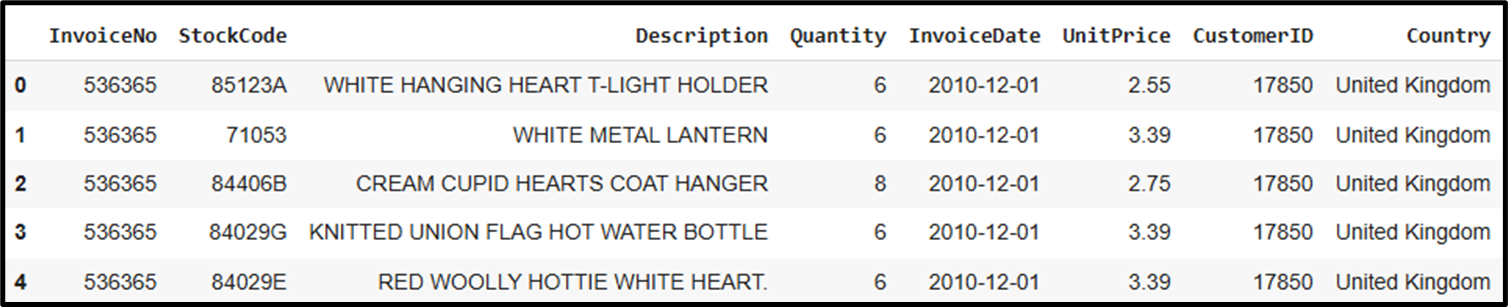


Figure 1: First five rows of "Online Retail" dataset, presented as a Pandas DataFrame

Throughout the project, the data was cleaned and transformed for use in clustering and classification tasks. The data was clustered in order to group customers in terms of their valuableness. Models considered for use in the clustering task include K-Means, Bisecting K-Means, Latent Dirichlet allocation, and Gaussian Mixture Model. The classification task similarly aimed to predict customer value given clustering results, and handle class imbalances and determine feature importance. Some of the models considered for classification include Logistic Regression, Random Forest Classifier, Decision Tree Classifier, and Naive Bayes models.

**Methodology**

Much of the initial data processing was inspired by the Databricks Solution Accelerator “Analyze Customer Lifetime Value.” The initial cleaning procedures and exploratory analysis of the Solution Accelerator served as guidance for the steps completed in this project.

The first component of the project’s methods included processing and scrubbing the data. Overall, the dataset was mostly clean, but there were several issues that needed correction for the success of this project. Using PySpark’s WithColumn function, a new column was created to describe the sales amount for each row, defined as the Quantity of the item times the Unit Price of the item. The creation of this column ensured that each item and its cost was properly accounted for in each row. Also, the InvoiceDate column had to be changed from a “string” data type to a “date” data type. Therefore, each purchase could be represented by its date of purchase.

Following the creation of the Sales column, some exploratory analysis was completed. Some exploration included information about daily transactions, including the customer, the date of purchase, the transaction, and the cost of the transaction. From the pivot table created using Spark SQL, visualizations regarding the daily sales were able to be created.

Also from the daily transactions, some scrubbing of the data was completed. By ordering the data in order of largest daily transactions, some customers had purchases of over £70,000, which seemed unrealistic compared to the rest of the data. Therefore, customers with extremely large daily purchases were removed using the Filter function in PySpark.

Continuing with the exploration of data, monthly sales reports were also generated. Upon doing this, it was evident that the December 2011 were significantly lower than every other month. With further inspection, it was clear that December 2011 only contained 3 days of data, while every other month was almost entirely included. Therefore, for consistency purposes, the data was limited to December 1, 2010 to November 30, 2011 to include exactly a full year’s worth of data at the company. A graph of a graph of a bar

Description automatically generated with medium confidence

Figure 2: Monthly Transactions Summary over the Year, Colored by Sales Amount

Lastly, as the project aims to highlight the company’s most valuable customers, rows with null CustomerIDs were removed using PySpark’s Filter function. Since these purchases could not be traced back to a given customer, they were unsupportive in the goal of the project.

After processing the data, the data had to be transformed to prepare for modeling. The first portion of the preparation was to create a new DataFrame containing customer information. The new DataFrame began with only distinct customer IDs. Then, the recency, frequency, and monetary value (RFM) values were computed for each distinct customer. The recency of the customer was calculated by finding the “current date” of the data, or the maximum of the invoice dates, November 30, 2011. Then, the last purchase date was found for each unique customer, and the two quantities were subtracted to find the number of days since the last purchase. The frequency for each customer was calculated by aggregating the data to count the number of unique invoice numbers. Lastly, the monetary value was calculated by computing the sum of the sales amount for each customer. These metrics were then joined to the customers data to get a complete picture of the customer’s ID, recency, frequency, and monetary value. Further cleaning of this data included removing customers with negative monetary values. The Databricks Solution Accelerator calculated the same RFM metrics, as well as age, but used slightly different methodology. After the calculation of metrics, the remainder of the methodology, especially pertaining to predictions and classification, differs from that of which was used in the Databricks Solution Accelerator.

For clustering tasks, the dataset needs to be scaled. Therefore, the customer RFM values were scaled using the MinMaxScaler in PySpark and were returned as a new columns in the DataFrame. Additionally, PySpark’s ML models require the data to include a vector of its features, so vectors of both the unscaled data (for use in classification) and scaled data (for use in clustering) were created using PySpark’s VectorAssembler. Lastly, the data was split into train and test datasets, randomly splitting into 70% and 30% of the data.

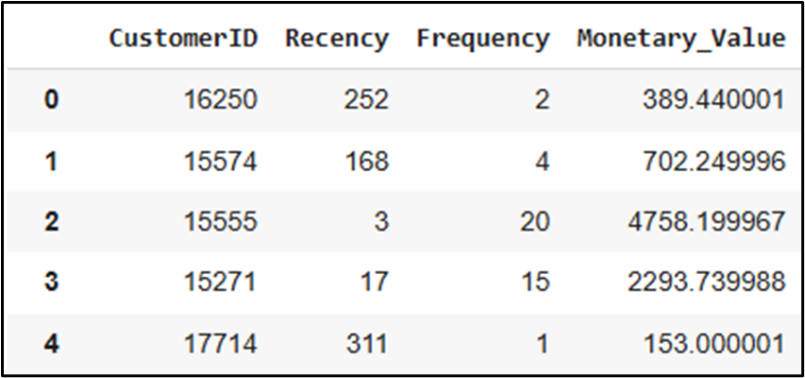


Figure 3: Customers Table, including their Calculated RFM Metrics

Once the dataset was completely transformed, machine learning modeling was underway. The project began with clustering the data in order to obtain labels for “valuable” and “non valuable” groups. The models used were the K-Means clustering model, and the Bisecting K-Means model from the PySpark Machine Learning models. The models were each created by searching for the features vector, and then fitted using the train data and transformed using the test data. The train data contained 2982 customers while the test data consisted of 1294 customers. The clustering models were each built using their default parameters, and information about how to apply the clustering models were obtained from the Drabas & Lee textbook (2017) as well as the *Clustering* documentation from Apache Spark.

After applying the dataset to the clustering task, predictions of clusters were created for the test data set. The data was grouped into clusters 0 and 1, and each cluster represented a group of more valuable customers and a group of less valuable customers. Using this labeled dataset, the classification models can be applied to train the model to predict a customer’s valuableness based on their recency, frequency, and monetary value. The classification model taken as the baseline was the Logistic Regression model, and the alternative model was taken as the Random Forest Classifier. These models were built based on the test data from the K-Means clustering results. To train and test the classification models, the test data results from clustering was split into 70% train and 30% test, at 926 customers (691 valuable) and 368 customers (281 valuable), respectively. The Logistic Regression models were built using their default parameters and were prepared using information from the Drabas & Lee textbook (2017) and the *Classification and Regression* documentation from Apache Spark. Each model was built with its default parameters. The Logistic Regression default parameters include max iterations of 100, regParam of 0.0, elasticNetParam of 0.0, and threshold of 0.5. The Random Forest Classifier default parameters include numTrees of 20, maxDepth of 5, and maxBins of 32. Each model included specification to identify the features vector column, and the clustering labels column. Each model was run and compared independently.

**Evaluation**

The clustering models were evaluated based primarily on their Silhouette value. The Silhouette score describes how well cohesive the clusters are in comparison to the other clusters, so it is a strong measure of how well grouped the clusters are. The K-Means model contained a Silhouette value of 0.858, and similarly, the bisecting K-Means model had a Silhouette value of 0.858. Some differences mostly came from each model’s run time and susceptibility to errors. The K-Means clustering typically ran around 55.29 seconds, while the Bisecting K-Means clustering ran faster at 29.93 seconds. However, when running the two models, the Bisecting K-Means model frequently would encounter run failures with limited explanation. Because the two models have the same silhouette score and reasonably similar run times, the K-Means model was preferred over the Bisecting K-Means for its consistency in being able to run.

The classification models were evaluated based on several measures, including precision (positive predictive value), recall (true positive rate), F-1 score (average of precision and recall), and support (number of incidences in each class). The baseline model, Logistic Regression, had precision, recall, and F-1 scores of 1.000, and had 81 and 287 for support of cluster 0 and 1. The alternative model, Random Forest Classifier, had precision of 0.993, recall of 1.000, and F-1 of 0.996, and had 81 and 287 for support of cluster 0 and 1. The documentation for computing each of these evaluation measures was *Evaluation Metrics – RDD-based API* from the Apache Spark documentation site. Based on these evaluation measures, the Logistic Regression model slightly outperformed the Random Forest Classifier, but overall, the two models were very accurate. Comparing the models by their run times, the Logistic Regression model ran in 49.14 seconds while the Random Forest Classifier ran in 47.09 seconds, so the Random Forest Classifier was just slightly faster to run. Comparing the two models’ confusion matrices, the Logistic Regression model identified zero false positives or negatives out of 368 customers, and the Random Forest Classifier model identified two false positives out of 368 customers. Thus, each model had minimal misclassifications, but the Logistic Regression model was able to correctly classify all customers.

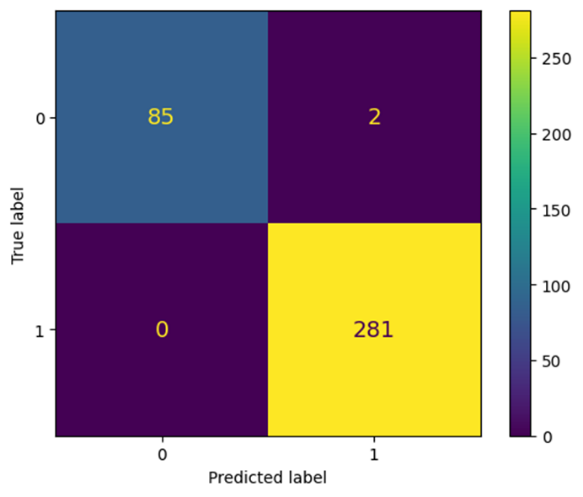
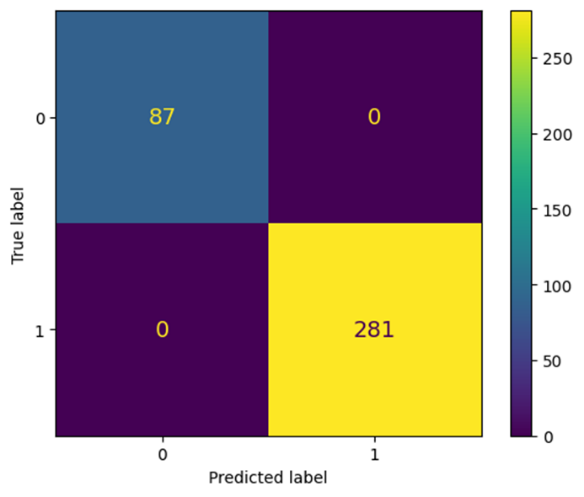


Figure 4: Confusion Matrix for Logistic Regression Model

Figure 5: Confusion Matrix for Random Forest Classifier

Following the creation of the initial models, the models were used in 3-fold cross validation to compare the two models to one another. Using the 3-folds to compute the F1 score of the two models, the following results were acquired.

Table 1: F1 Scores computed using 3-fold Cross Validation

|  |  |
| --- | --- |
| Logistic Regression Model | Random Forest Classifier Model |
| 1.0000 | 0.9980 |
| 1.0000 | 0.9955 |
| 1.0000 | 1.0000 |

The null hypothesis of the hypothesis test is that the alternative model, Random Forest Classifier, does not outperform the baseline model, Logistic Regression. Therefore, the alternative hypothesis claims that the alternative model, Random Forest Classifier, outperforms the baseline model, Logistic Regression. Using these F1 score results from the cross validation, a p-value was obtained as 0.239. As this value is greater than 0.05, the null hypothesis cannot be rejected at the 95% confidence level. Therefore, the two models do not outperform one another.

**Conclusion and Discussion**

Based on evaluation measures, the clustering PySpark ML models acted very similarly, however, the K-Means model was utilized for its labels in the classification portion of the project as the model was consistently reliable and rarely caused run time failures, unlike the Bisecting K-Means model. Therefore, the K-Means model was preferred for clustering.

Following the classification of the dataset using the labels from the K-Means clustering, the Logistic Regression and Random Forest Classifier models were both quite accurate, but the Logistic Regression model was slightly more accurate than the Random Forest Classifier, correctly identifying all valuable and nonvaluable customers with precision, recall, and F-1 scores of 1.00. The Logistic Regression model correctly predicted all 368 of 368 customers, while the Random Forest Classifier model correctly predicted 366 of 368 customers. For its high accuracy in predicting customer valuableness. From using K-fold cross validation of the models, the models were determined to not have higher performance than one another. Therefore, the alternative model, Random Forest Classifier, does not outperform the baseline Logistic Regression model.

A graph of a customer

Description automatically generated with medium confidenceAlthough having high evaluation measures is sought after to verify the goodness of fit of a model, there are some questions raised about the extremely high measures for the classification models. Since the models were trained based on labels from clustering models, it is questionable if the accuracy is inflated because of the cooperation between the two tasks. Additionally, there was limited data available for the classification tasks after splitting and testing the data several time. Therefore, the model may appear to be over precise with a small dataset available.

From the clustering and classification tasks, conclusions can be drawn regarding a customer’s value based on their recency, frequency, and monetary value scores. Each clustering model concluded that customers with low recency, high frequency, and high monetary value were the most valuable customers. On average, customers in this cluster had recency values of about 40.0 days, frequency values of about 5.8, and monetary value of 1809.1. Therefore, the most valuable customers tend to have shopped at the online store more recently, shop around 6 times in the year, and spent large amounts of money. The less valuable customers tended to have higher recency (averaging 234 days ago), lower frequency (average of 1.9 times a year), and lower monetary value (477.4 average spending amount).

Figure : Average RFM values per Cluster

A graph with blue and red dots

Description automatically generatedFrom the Logistic Regression classification model, the model appeared to rank the features of importance by their recency, then frequency, and lastly, the monetary value. This idea goes against some elementary thinking about marketing techniques, as many marketers may believe that how much money a customer spends determines how much value they have to the company. However, the Logistic Regression model argues that instead, focusing on how long it has been since a customer has shopped and how often a customer makes purchases are key indicators of a customer’s loyalty and shopping tendencies to a company.

Figure : 3D depiction of Clusters based on RFM Values

This project presented itself with many difficulties. Some of the major difficulties consisted of learning about and applying the ML models. I have limited experience with machine learning and have only used it in one other course, which I was taking concurrently with this one. Also, the modeling I had done was in Python, so I had to learn how to apply it in PySpark.

Some opportunities of learning data analytics within this project include the importance of understanding the dataset. At first glance of the dataset, discovering how to decipher a customer’s valuableness seems daunting. However, upon exploration of the data, various metrics about customers can be gathered for analysis. Additionally, this dataset did not explicitly define customers as valuable or non valuable, so the dataset was unlabeled. I had not worked with an unlabeled dataset before, so attempting to classify the data right away was not possible. Therefore, I had to perform clustering on the data before applying classification so that the classification models would have labels to predict based on. This project was a great experience for learning how to get comfortable with a dataset and how to approach unlabeled data.

If this project were continued in the future, there are many aspects I would like to further investigate. I would begin by finetuning the clustering and classification models to obtain different evaluation measures. Additionally, adjusting the number of clusters could create more groupings or tiers of customers. This would help marketers choose to mostly invest in their top tiers and avoid overspending on less valuable customers. This project would also be quite interesting to see given more data, either including two years of data, or an all time customer record to reduce how much the models are limited by small data amounts. Finally, I would work to come up with a numerical quantity for the Customer Lifetime Value (CLV) based on the RFM values. Therefore, the most valuable customers could be quantified and rewarded with special offers and discounts. Additionally, having CLVs for each customer could help discover where cutoffs lie in the clusters to determine which customers are valuable.

Bibliography

*Analyze Customer Lifetime Value*. Databricks. <https://www.databricks.com/solutions/accelerators/customer-lifetime-value>

Chen, D. (2015, November 5). *Online Retail*. UC Irvine Machine Learning Repository. <https://archive.ics.uci.edu/dataset/352/online+retail>

*Classification*. Apache Spark. <https://spark.apache.org/docs/latest/ml-classification-regression.html>

*Clustering.* Apache Spark. <https://spark.apache.org/docs/latest/ml-clustering.html>

Drabas, T., & Lee, D. (2017)**. *Learning PySpark*.** Birmingham, UK: Packt Publishing.

*Evaluation Metrics – RDD-based API*. Apache Spark. <https://spark.apache.org/docs/3.5.3/mllib-evaluation-metrics.html>

Murphy, C. (2024, August 14). *What Is Recency, Frequency, Monetary Value (RFM) in Marketing?.* Investopedia. <https://www.investopedia.com/terms/r/rfm-recency-frequency-monetary-value.asp>

Saker, R. et al. (2020, June 3). *Customer Lifetime Value Part 1: Estimating Customer Lifetimes.* Databricks. <https://www.databricks.com/blog/2020/06/03/customer-lifetime-value-part-1-estimating-customer-lifetimes.html>