The ABC Bicycle Company

Capstone Project by Helen Sew
10.04.2023

Executive Summary

This report provides an analysis of customer spending habits and identifies important customer groups using RFM analysis. The report also predicts the lifetime value of each customer and identifies key factors that influence it through a regression model. The analysis further explores the important features of the best model and examines the demographics of the customers based on those features. While the report provides a brief product analysis by product name, further research is needed to better understand the specific product preferences of each customer cluster.

After performing the data analysis, the following conclusions were reached:

- Customers in cluster 0 have a higher yearly income, education level, and car ownership rate compared to customers in cluster 2. Both clusters have a similar gender distribution and occupation category, and the majority of customers in both clusters do not have children.

- Customers in cluster 0 have a higher mean for monetary value, frequency, total purchase YTD, and lifetime value compared to customers in cluster 2.

- The company could tailor its marketing strategies to better suit the demographic characteristics of each cluster. For cluster 0, the company could offer more premium products or services, while for cluster 2, they could focus on affordability and value. Targeted advertising on social media platforms or email campaigns could be used to reach out to each cluster more effectively.

- If the company is constrained by budget or resources, it would be recommended to focus on cluster 0 first since these customers generate more revenue and have a higher lifetime value. Once the company has successfully targeted and acquired more customers in cluster 0, they can then shift their focus to cluster 2.

Table of Contents

Introduction

The objective of this assignment is to analyze the data collected by The ABC Bicycle Company about their customers, which includes personal and contact details along with their purchase history. The main goals are to gain insights into customer spending habits, identify target customer groups, and predict the lifetime value of each customer.

In this report the datasets described below were examined using data analytics tools including: codes written in Python language, MS Excel and Power BI software.

About the datasets:

The file aw_bicycle.xlsx (in Block 5 data folder) contains a few sheets:

- Sales Customer: connecting file between customer and sales
- Sales SalesOrderHeader: information regarding sales orders (order/invoice level)
- Sales SalesOrderDetail: information regarding each sales item (itemized level)
- Sales vPersonDemographics: customer attributes (e.g. birthday, marital status, gender, etc.)
- Production Product: product name and attributes

To connect the datasets, here are the key columns/variables:

- Sales Customer to Sales SalesOrderHeader: CustomerID
- Sales Customer to Sales vPersonDemographics: PersonID to BusinessEntityID
- Sales SalesOrderHeader to Sales SalesOrderDetail: SalesOrderID
- Sales SalesOrderDetail to Production Product: ProductID

Preliminary Data Exploration

The preliminary data analysis was performed to study the dataset, identify patterns, and clean data to ensure that it is accurate and reliable. In this scenario, we are dealing with two dataframes: the Sales Order Details dataframe and the Sales Order Header data frame. The Sales Order Details dataframe contains information about the products sold in each order, while the Sales Order Header data frame contains information about the orders themselves. There are a total of 121, 317 sales transactions.

Methodology:
The data exploration process involved several steps. First, the two dataframes were merged using the Sales Order ID column. Next, duplicates were identified and removed using the drop_duplicates() function in Pandas. Missing values were then identified using the isnal() function, and appropriate measures were taken to either remove the rows containing missing values or impute the missing values with a suitable value. Finally, type conversion was performed using the astype() function in Pandas to ensure that each column had the correct data type.

Results:
After performing the above data exploration methodology mentioned above, the following key statements can be drawn:

- There are no duplicates and hence each observation is unique.
- There are missing values in the merged dataframe, which were subsequently removed.
- Convert following variables to categorical features.

    - SalesOrderID
    - RevisionNumber (2 unique number i.e. 8 & 9
    - Status (1 unique number i.e. 5)
    - CustomerID
    - TerritoryID
    - BillToAddressID
    - ShipToAddressID
    - ShipMethodID
    - SalesOrderDetailID
    - ProductID
    - SpecialOfferID

By conducting these steps in preliminary data exploration, we can ensure that the data is clean, accurate, and reliable before proceeding with RFM analysis.

RFM Clustering Analysis

In order to segment customers based on relevant business objectives, such as market share and profit margin, we need to engineer retail features that directly influence these objectives. Specifically, we will focus on three features: Recency (R), Frequency (F), and Monetary (M).

Recency measures the number of days since a customer's last purchase, and is an important indicator of customer loyalty and repeat frequency. Frequency measures the number of transactions made by a customer, which is a measure of product demand. Monetary measures the total revenue contributed by each customer.

RFM analysis is a data-driven customer segmentation technique that utilizes historical customer data to identify the most valuable customers for a business. By analyzing these three dimensions, RFM analysis enables businesses to segment their customers into groups based on their behavior and tailor their marketing strategies to each group accordingly. This helps businesses to better understand their customers and improve their overall profitability.

RFM Calculation

Step 1: Compute Recency

```
#### create recency variable
# based on latest OrderDate as days = 0 (max OrderDate - current line
OrderDate)

df_sales_order_detail['recency'] = max(df_sales_order_detail.OrderDate) -
df_sales_order_detail.OrderDate
```

Step 2: Compute Frequency

```
rfm_f =
df_sales_order_detail.groupby('CustomerID')['SalesOrderID'].count()
rfm_f
```

Step 3: Compute Monetary

```
#### create monetary variable
# Monetary metric is simply total revenue generated by each customer
# use LineTotal (can be computed by quantity * unit_price)
# LineTotal is the total amount for each line item
# sum LineTotal, groupby CustomerID

rfm_m =
df_sales_order_detail.groupby('CustomerID')['LineTotal'].sum().round(2)
rfm_m
```

Step 4: Merge - we combine recency, frequency and monetary dataframe together so that we have the RFM metrics per customer.

```
# combine recency, frequency, monetary datasets together
# makign sure there are no missing values
# CustomerID should be the index, 'recency', 'frequency, 'monetary' should
be the columns
# We use the concat function so that we can merge all 3 dataframes in 1
pass.

rfm = pd.concat([rfm_r, rfm_f, rfm_m], axis=1, join='inner')
rfm

#Lastly, since we require all rows to have values, we user inner join
which will drop any customer id that have missing value
#for either R, F, M.
```

CustomerID	recency	frequency	monetary

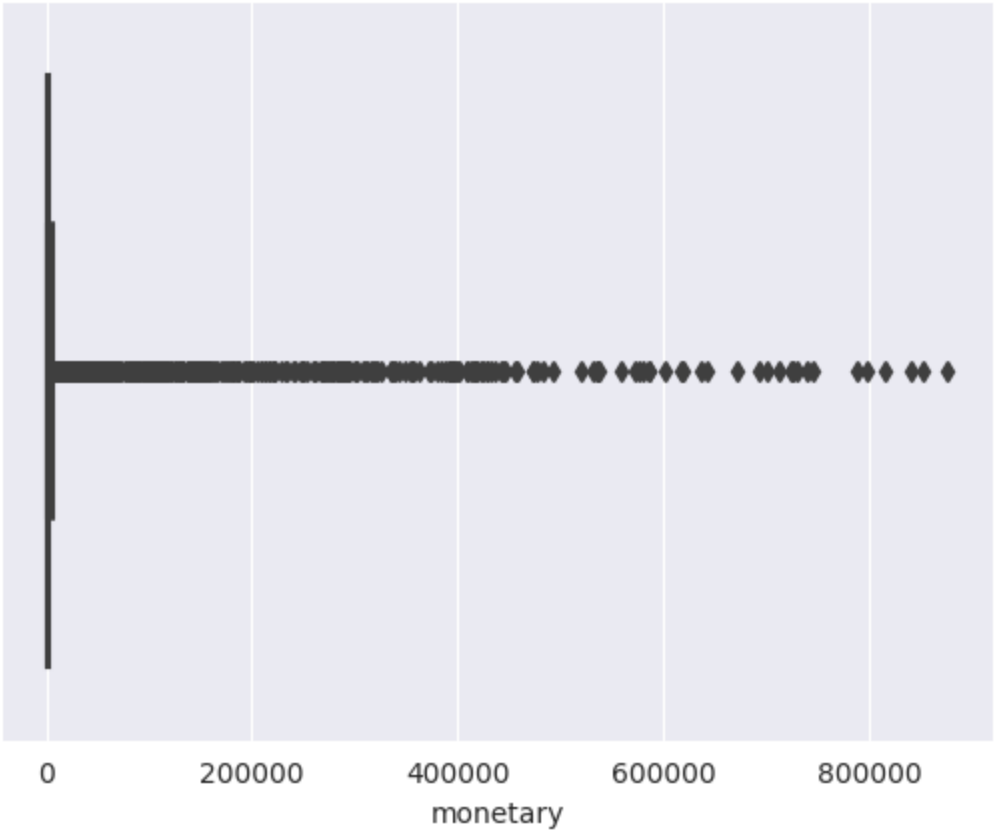
11000	270	8	8248.99
11001	49	11	6383.88
11002	339	4	8114.04
11003	263	9	8139.29
11004	272	6	8196.01
...	...	...	...
30114	121	30	11652.99
30115	91	21	8917.56
30116	121	119	187114.20
30117	91	436	816755.58
30118	60	289	278568.57

19119 rows × 3 columns

Step 5: Inspect the dataset

```
rfm.describe()
```

```
#checking outliers on 'monetary' column
#the min value is 1.37 and max value is 877,107.19 suggests that there
#may be one or more outliers in the dataset.
# plot boxplot of 'monetary'

sns.boxplot(x=rfm['monetary'], orient='h')
```



Outlier Removal

Remove outliers - trim at 1st and 99th percentile

```
#### removing outliers
# trim the dataset at 1 percentile and 99th percentile
# which means removing records below 1st percentile and above 99th
percentile
# first, establish the 1st and 99th percentile
p1 = rfm.monetary.quantile(0.01)
p99 = rfm.monetary.quantile(0.99)

# Check the descriptive statistics to see if the numbers are better
(especially monetary)
rfm[(rfm.monetary >= p1) & (rfm.monetary <= p99)].describe()
```

Clustering Analysis

Clustering analysis is an unsupervised machine learning technique that involves grouping similar data points or objects based on their characteristics or features._The goal of clustering analysis is to divide a dataset into distinct groups, or clusters, in a way that maximizes the similarity within each cluster and minimizes the similarity between clusters.

Once the RFM analysis has been performed, the k-means clustering algorithm is utilized for customer segmentation. The following is a step-by-step guide for conducting clustering analysis using the k-means algorithm after RFM analysis:

Step 1: Setup

```
# run clustering using pycaret
from pycaret.clustering import *
```

```
model = setup(data=rfm,
              normalize=True,
              normalize_method='zscore',
              session_id=75)
```

Standardize the variables: Before applying the k-means algorithm, it is essential to standardize the variables to ensure that all variables contribute equally to the analysis. Standardization involves transforming the variables to have a mean of 0 and a standard deviation of 1.

Step 2 : Choose the number of clusters (K)

```
# run through multiple clusters to check which has the best results
# create empty dataframe to store results
results = pd.DataFrame()

#tqdm.trange allows us to see a progress bar when we use range function

from tqdm import trange

for i in trange(2, 11):
   kmeans = create_model('kmeans',
                         num_clusters=i,
                         init='k-means++',
                         n_init=10,
                         max_iter=300,
                         random_state=75
```

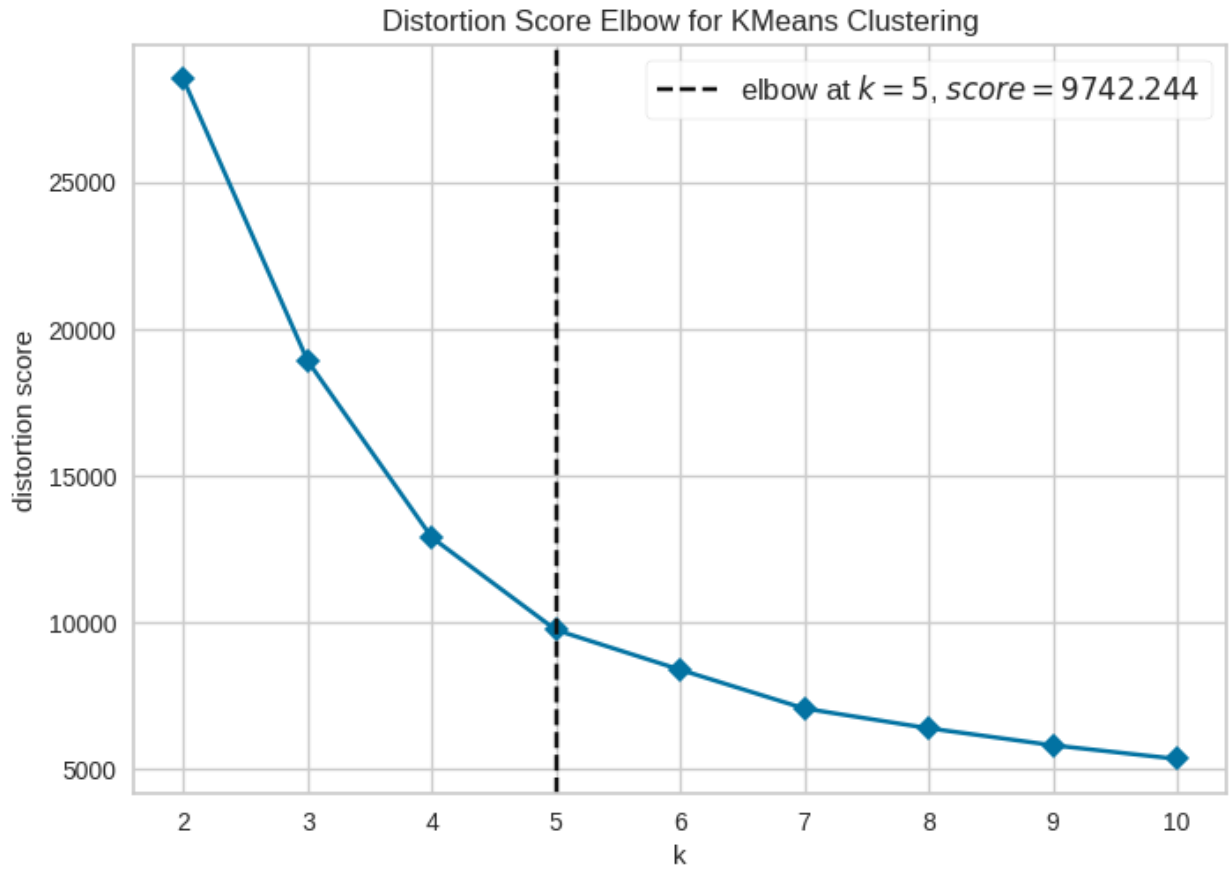
```
                          )
    metrics = pull()  # Extract results table into dataframe
    metrics['algo'] = 'kmeans'
    metrics['num_clusters'] = i
    results = results.append(metrics)
```

```
# check the results of mulitiple number of clusters

results

# Silhoutte says 2 clusters
# Calinski says 5 or more clusteres
# Davis says 2 clusters (smaller is better)
# plot model says 4
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness	algo	num_clusters
0	0.9144	18535.0406	0.4487	0	0	0	kmeans	2
0	0.4706	18745.3651	0.6844	0	0	0	kmeans	3
0	0.4829	21268.6839	0.6080	0	0	0	kmeans	4
0	0.4901	22638.4052	0.7135	0	0	0	kmeans	5
0	0.4943	21643.9463	0.7699	0	0	0	kmeans	6
0	0.3929	21995.3865	0.8383	0	0	0	kmeans	7
0	0.3926	21122.8836	0.8733	0	0	0	kmeans	8
0	0.3931	20556.3562	0.9195	0	0	0	kmeans	9
0	0.3935	20053.0562	0.8995	0	0	0	kmeans	10

Based on the table, it appears that the Silhouette score is highest for 2 clusters, while the Calinski-Harabasz score is highest for 5 clusters. The Davies-Bouldin score is lowest for 2 clusters, indicating that this may be the optimal number of clusters for the given data. However, it is also important to consider other factors such as the domain knowledge and the purpose of the clustering analysis before deciding on the optimal number of clusters. Therefore, further analysis is done using the elbow plot.

```
# plot the elbow plot to confirm 4 is recommended
plot_model(best_model, plot='elbow')
```



So how many clusters should we use? The practical consideration here is to balance between sufficient and overwhelming diversity.

I am inclined to choose 4 clusters.

Step 3 - assign cluster numbers to the rfm_df

```
# let's go with 4 clusters
# create model based on 4 clusters

best_model = create_model('kmeans',
                          num_clusters=4,
                          init='k-means++',
                          n_init=10,
                          max_iter=300,
                          random_state=75
```

```
)
```

	CustomerID	recency	frequency	monetary	Cluster
0	11000	270	8	8248.990234	Cluster 2
1	11001	49	11	6383.879883	Cluster 0
2	11002	339	4	8114.040039	Cluster 2
3	11003	263	9	8139.290039	Cluster 2
4	11004	272	6	8196.009766	Cluster 2
...	...	...	...	...	...
18846	30106	426	114	101203.578125	Cluster 1
18847	30108	457	105	144602.578125	Cluster 1
18848	30110	426	6	1625.280029	Cluster 2
18849	30114	121	30	11652.990234	Cluster 0
18850	30115	91	21	8917.559570	Cluster 0

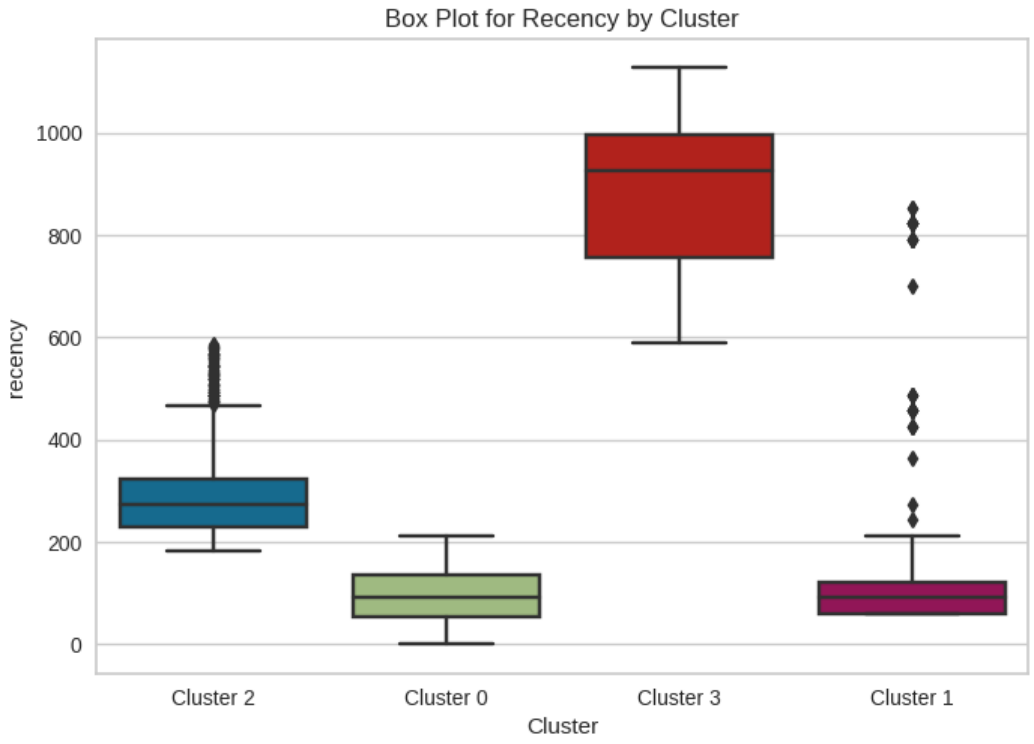
18851 rows × 5 columns

```
[ ]  1  rfm_df.Cluster.value_counts()
```

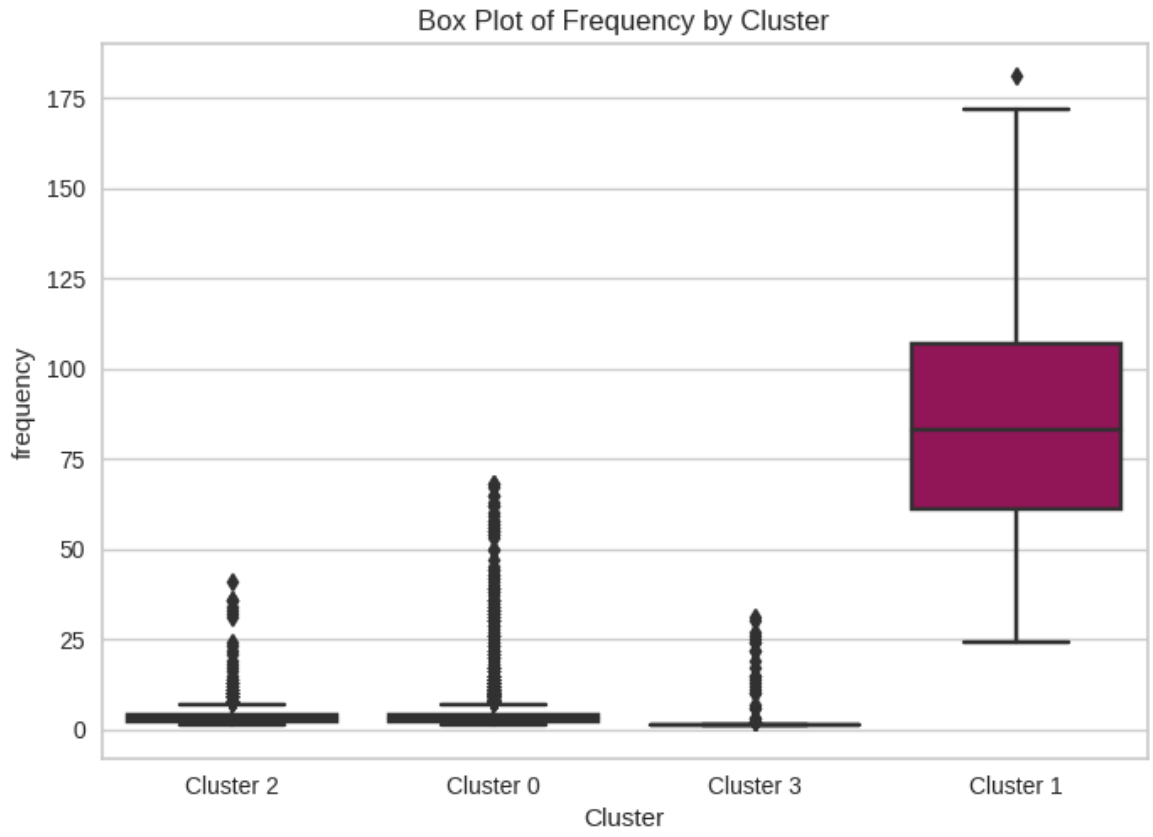
```
Cluster 0    10398
Cluster 2     7920
Cluster 3      392
Cluster 1      141
Name: Cluster, dtype: int64
```

Step 4: Inspecting the results

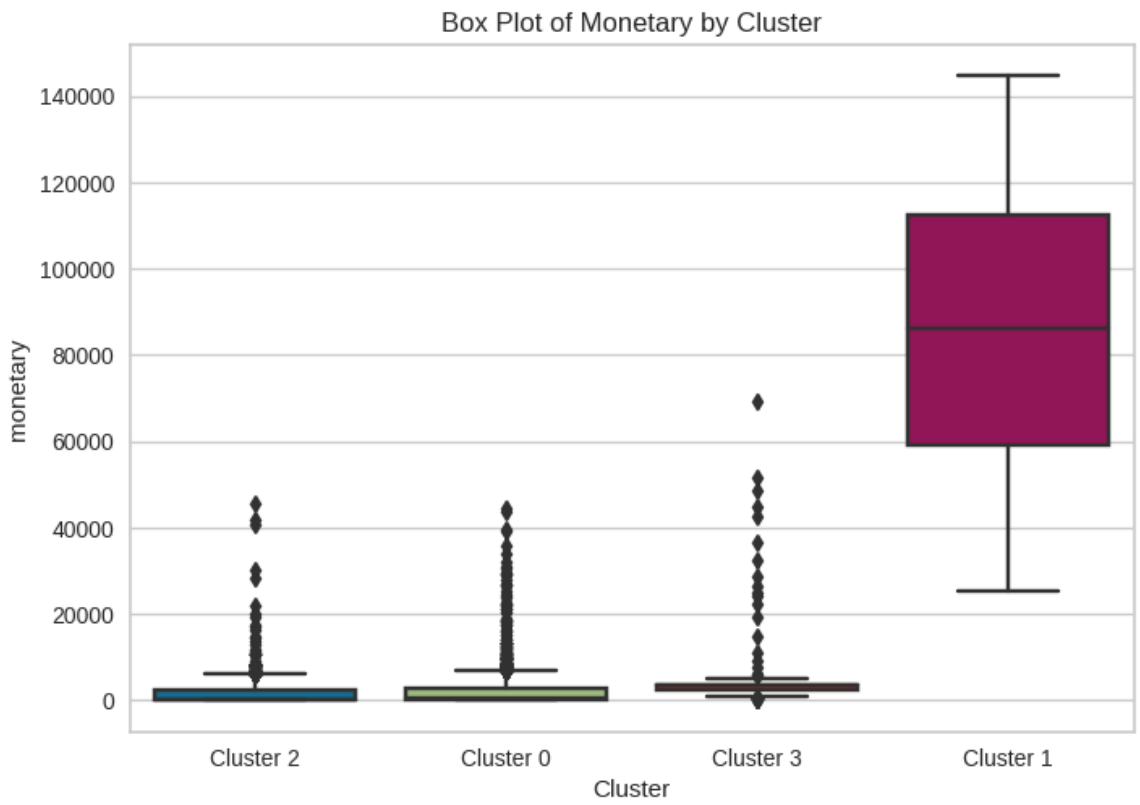
```
# Clusters vs Recency

#create a box plot of 'recency' for each cluster
sns.boxplot(x='Cluster', y='recency', data=rfm_df)

plt.title("Box Plot for Recency by Cluster")
```


Box Plot for Recency by Cluster
recency
1000
800
600
400
200
0
Cluster 2
Cluster 0
Cluster 3
Cluster 1
Cluster

```
# Clusters vs Frequency

#create a box plot of 'frequency' for each cluster
sns.boxplot(x='Cluster', y='frequency', data=rfm_df)

plt.title("Box Plot of Frequency by Cluster")
```




```
# Clusters vs Monetary Value

#create a box plot of 'monetary' for each cluster
sns.boxplot(x='Cluster', y='monetary', data=rfm_df)

plt.title("Box Plot of Monetary by Cluster")
```



Results and recommendations

Which cluster of customers are the most important?

Based on the RFM clustering analysis, the bicycle company should focus on cluster 1, which consists of high-value customers with high recency, high frequency, and high monetary value. This cluster represents loyal customers who have a high potential for repeat business, and it is important to retain them by providing excellent customer service, targeted marketing, and loyalty rewards programs. However, the impact of targeting this cluster may be limited due to its small size.

Cluster 0 and 2, with high recency, low frequency, and low monetary value, represent customers who have made fewer purchases and spend less money. Although these customers may not be as profitable as those in cluster 1, they still represent a significant customer base, and it is important to engage with them to encourage them to make more purchases. The bicycle company can use personalized outreach, special promotions or discounts, improved customer experience, and product recommendations to create a more personalized and targeted experience that speaks to their specific needs and interests.

To decide whether to focus on cluster 0 or 2, the company can compare the sizes of the clusters and look at other metrics that are relevant to their business goals, such as customer lifetime value or customer acquisition cost. By developing strategies to retain high-value customers in cluster 1 and engage with customers in cluster 0 or 2, the bicycle company can increase customer loyalty and profitability over time.

Customer Lifetime Value Prediction

Calculate lifetime value per customer and add a new column in rfm_df

```
# calculate lifetime value
# average spend value * years of service
rfm_df['lifetime_value'] = ave_spend_value * years_of_service
rfm_df
```

	CustomerID	recency	frequency	monetary	Cluster	lifetime_value
0	11000	270	8	8248.990234	Cluster 2	8.826420e+05
1	11001	49	11	6383.879883	Cluster 0	6.250399e+05
2	11002	339	4	8114.040039	Cluster 2	1.596437e+06
3	11003	263	9	8139.290039	Cluster 2	7.804675e+05
4	11004	272	6	8196.009766	Cluster 2	1.166565e+06
...	...	...	...	...	...	...
18846	30106	426	114	101203.578125	Cluster 1	6.214255e+05
18847	30108	457	105	144602.578125	Cluster 1	9.213250e+05
18848	30110	426	6	1625.280029	Cluster 2	1.896160e+05
18849	30114	121	30	11652.990234	Cluster 0	3.903752e+05
18850	30115	91	21	8917.559570	Cluster 0	4.395083e+05

18851 rows × 6 columns

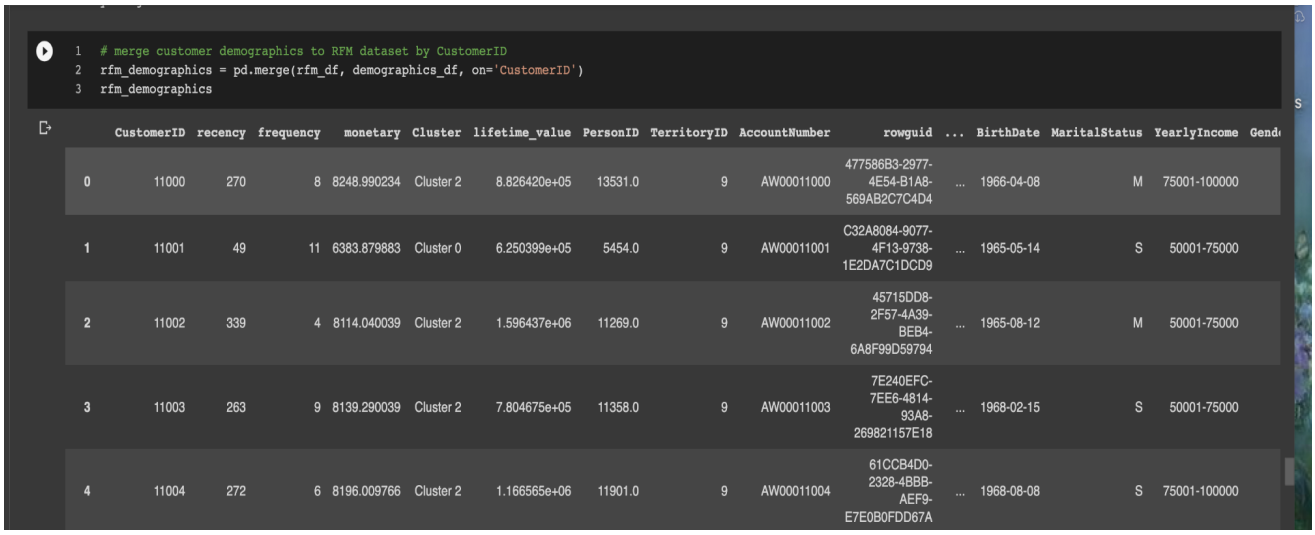
Get in customer demographics

```
# merge to SalesCustomer to get CustomerID into Person Demographics
demographics_df = pd.merge(df_sales_customer, df_demograhics, left_on='PersonID', right_on='BusinessEntityID')
demographics_df
```

	CustomerID	PersonID	StoreID	TerritoryID	AccountNumber	rowguid	ModifiedDate	BusinessEntityID	TotalPurchaseYTD	DateFirstPurchase	BirthDate	MaritalStatus	Ye
0	11015	10963.0	NaN	4	AW00011015	F791BD74-EB82-4631-B9FC-F9FEE621FD13	2014-09-12 11:15:07	10963	2182.9700	2003-07-22	1979-02-27	S	
1	11016	3800.0	NaN	4	AW00011016	023843CA-25FB-42BF-AC37-FAF6F4120DAC	2014-09-12 11:15:07	3800	2327.7000	2003-08-13	1979-04-28	M	
2	11023	4373.0	NaN	4	AW00011023	A2833BD8-44A8-4665-808E-33E19ECD7F54	2014-09-12 11:15:07	4373	63.6700	2003-08-21	1978-10-11	M	
3	11024	16843.0	NaN	4	AW00011024	EE8EEBF8-117C-4027-873D-6113B0E97489	2014-09-12 11:15:07	16843	43.9800	2003-12-29	1978-09-17	M	
						2E40BEAF-							

Merge customer demographics data frame with rfm data frame

```
# merge customer demographics to RFM dataset by CustomerID
rfm_demographics = pd.merge(rfm_df, demographics_df, on='CustomerID')
rfm_demographics
```



	CustomerID	recency	frequency	monetary	Cluster	lifetime_value	PersonID	TerritoryID	AccountNumber	rowguid	...	BirthDate	MaritalStatus	YearlyIncome	Gend
0	11000	270	8	8248.990234	Cluster 2	8.826420e+05	13531.0	9	AW00011000	477586B3-2977-4E54-B1A8-569AB2C7C4D4	...	1966-04-08	M	75001-100000	
1	11001	49	11	6383.879883	Cluster 0	6.250399e+05	5454.0	9	AW00011001	C32A8084-9077-4F13-9738-1E2DA7C1DCD9	...	1965-05-14	S	50001-75000	
2	11002	339	4	8114.040039	Cluster 2	1.596437e+06	11269.0	9	AW00011002	45715DD8-2F57-4A39-BEB4-6A8F99D59794	...	1965-08-12	M	50001-75000	
3	11003	263	9	8139.290039	Cluster 2	7.804675e+05	11358.0	9	AW00011003	7E240EFC-7EE6-4814-93A8-269821157E18	...	1968-02-15	S	50001-75000	
4	11004	272	6	8196.009766	Cluster 2	1.166565e+06	11901.0	9	AW00011004	61CCB4D0-2328-4BBB-AEF9-E7E0B0FDD67A	...	1968-08-08	S	75001-100000	

Use regression analysis to predict lifetime_value per customer

Step 1: to understand the dataset i.e. rfm_demographics and set aside 10% as unseen data to be used for predictions.

```
rfm_demo_report = rfm_demographics.profile_report(minimal=True)
rfm_demo_report.to_file('profile_report_rfm_demongraphics.html')
rfm_demo_report
```

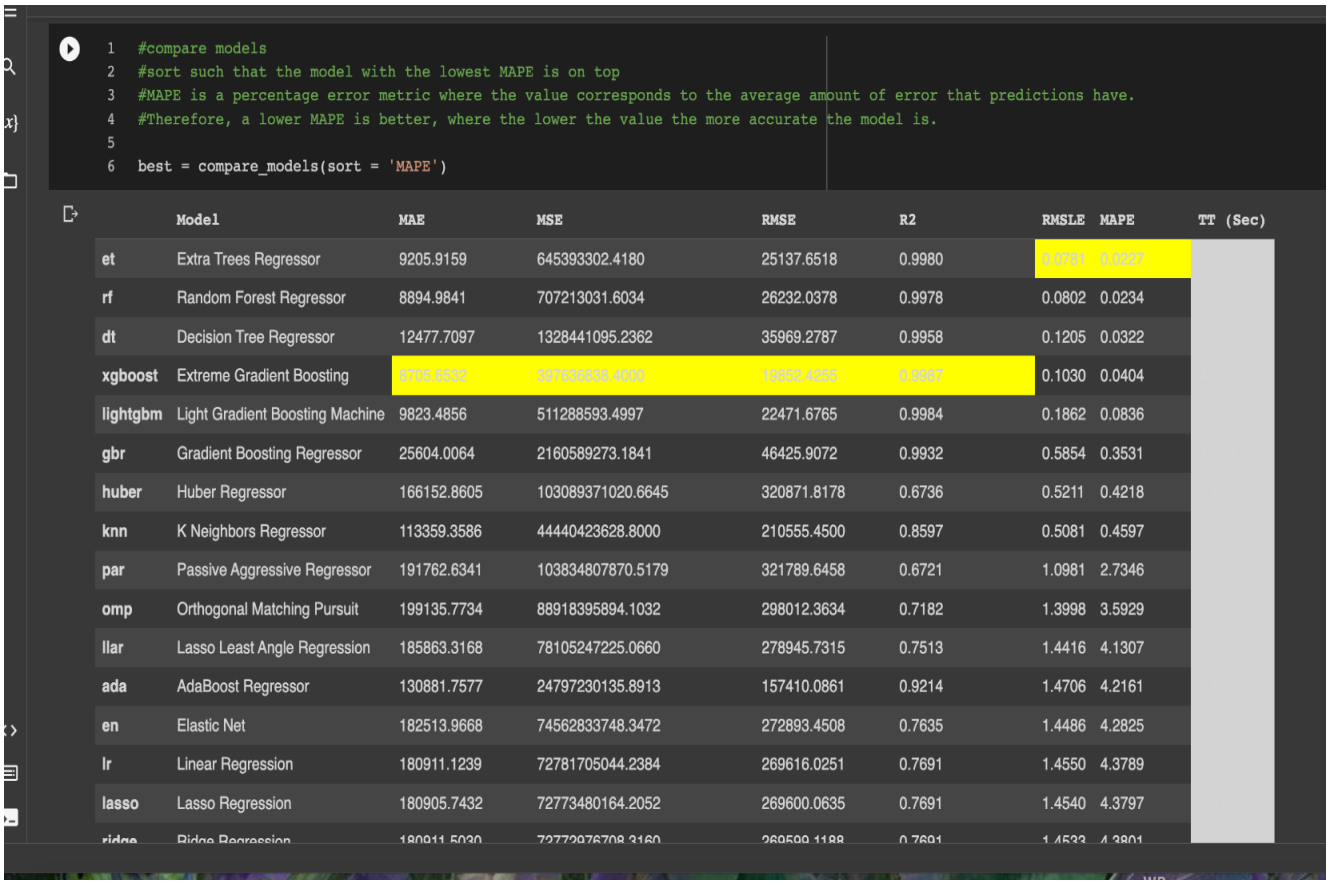
```
#10% of rfm_demographics data has been withheld for predictions (i.e.
predict_model)
rfm_demo_data = rfm_demographics.sample(frac=0.9, random_state=75)
data_unseen = rfm_demographics.drop(rfm_demo_data.index)

print('Data for Modeling: ' + str(rfm_demo_data.shape))
print('Unseen Data for Prediction: ' + str(data_unseen.shape))
```

Step 2 : Set up environment in PyCaret

```
from pycaret.regression import *

model = setup(
```

```
    data=rfm_demo_data,
    target='lifetime_value',
    session_id=75
    #categorical_features=cat_vars,
    #ignore_features=['CustomerID', 'AccountNumber', 'rowguid']

)
```

Step 3: Compare all models



```
#compare models
#sort such that the model with the lowest MAPE is on top
#MAPE is a percentage error metric where the value corresponds to the average amount of error that predictions have.
#Therefore, a lower MAPE is better, where the lower the value the more accurate the model is.

best = compare_models(sort = 'MAPE')
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
et	Extra Trees Regressor	9205.9159	645393302.4180	25137.6518	0.9980	0.0781	0.0227	
rf	Random Forest Regressor	8894.9841	707213031.6034	26232.0378	0.9978	0.0802	0.0234	
dt	Decision Tree Regressor	12477.7097	1328441095.2362	35969.2787	0.9958	0.1205	0.0322	
xgboost	Extreme Gradient Boosting	9705.6532	937036838.4000	30652.4255	0.9967	0.1030	0.0404	
lightgbm	Light Gradient Boosting Machine	9823.4856	511288593.4997	22471.6765	0.9984	0.1862	0.0836	
gbr	Gradient Boosting Regressor	25604.0064	2160589273.1841	46425.9072	0.9932	0.5854	0.3531	
huber	Huber Regressor	166152.8605	103089371020.6645	320871.8178	0.6736	0.5211	0.4218	
knn	K Neighbors Regressor	113359.3586	44440423628.8000	210555.4500	0.8597	0.5081	0.4597	
par	Passive Aggressive Regressor	191762.6341	103834807870.5179	321789.6458	0.6721	1.0981	2.7346	
omp	Orthogonal Matching Pursuit	199135.7734	88918395894.1032	298012.3634	0.7182	1.3998	3.5929	
llar	Lasso Least Angle Regression	185863.3168	78105247225.0660	278945.7315	0.7513	1.4416	4.1307	
ada	AdaBoost Regressor	130881.7577	24797230135.8913	157410.0861	0.9214	1.4706	4.2161	
en	Elastic Net	182513.9668	74562833748.3472	272893.4508	0.7635	1.4486	4.2825	
lr	Linear Regression	180911.1239	72781705044.2384	269616.0251	0.7691	1.4550	4.3789	
lasso	Lasso Regression	180905.7432	72773480164.2052	269600.0635	0.7691	1.4540	4.3797	
ridge	Ridge Regression	180911.5030	72772976708.3160	269599.1188	0.7691	1.4533	4.3801	

Step 4: Create a model

To work with the below models as our candidate models to find a best model:

- Random Forest Regressor (rf)
- Extra Trees Regressor (et)
- Extreme Gradient Boosting (xgboost)

```
compare_models(include=['rf', 'et','xgboost'],
               sort='MAPE'
               )
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
et	Extra Trees Regressor	9205.9159	645393302.4180	25137.6518	0.9980	0.0781	0.0227	
rf	Random Forest Regressor	8894.9841	707213031.6034	26232.0378	0.9978	0.0802	0.0234	
xgboost	Extreme Gradient Boosting	8705.6532	397636838.4000	19852.4255	0.9987	0.1030	0.0404	

```
ExtraTreesRegressor
ExtraTreesRegressor(n_jobs=-1, random_state=75)
```

We can tune the model using tune_model() to find the better result, however, we exclude this in this exercise due to limited computer resources and long execution times.

Choose the best model:

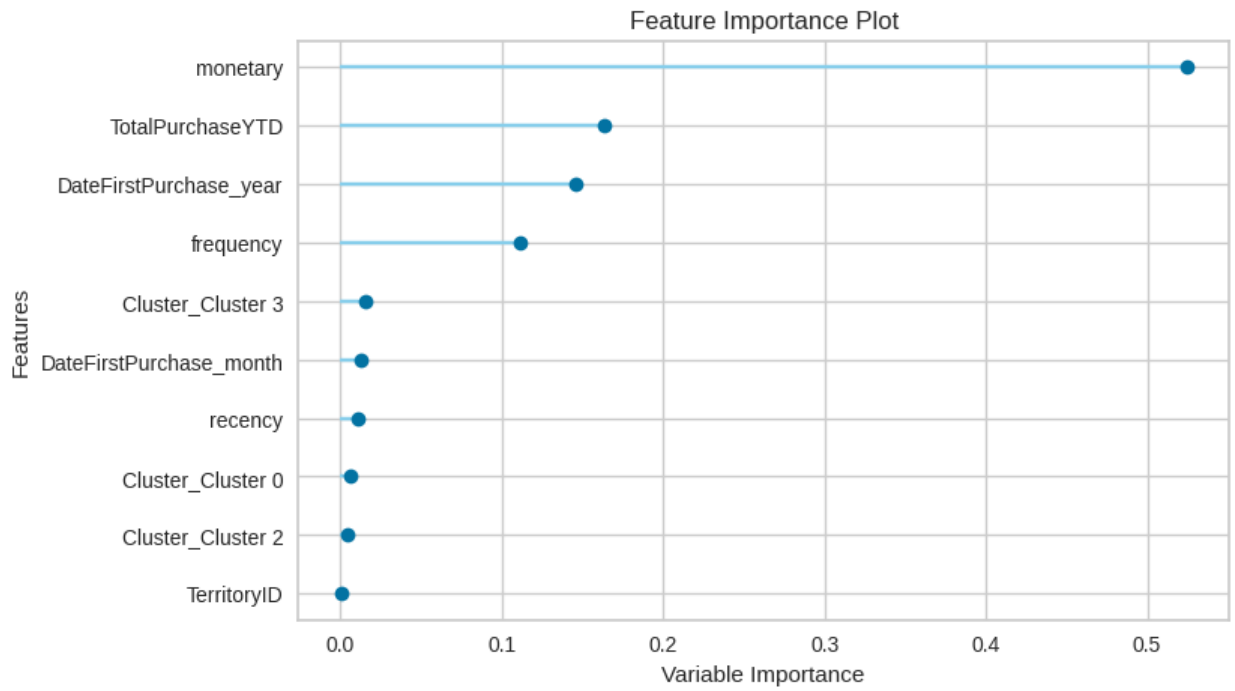
Metrics alone are not the only criteria we should consider when finalizing the best model for production. Other factors to consider include training time, the standard deviation of k-folds etc. For now, let's move forward considering the Extra Trees Regressor stored in the 'et'' variable as our best model.

```
# create the best model
best_model = et
```

Step 5: Plot a model

Feature Important Plot

```
plot_model(best_model, plot='feature')
```



Another way to analyze the performance of models is to use the evaluate_model

```
evaluate_model(best_model)
```

Step 6 Predict on Test/Hold-out sample

Before finalizing the model, it is advisable to perform one final check by predicting the test/hold set and reviewing the evaluation metrics. 30% (4971 samples) of the data has been separated out as a test sample. All of the evaluation metrics we have seen above are cross-validated results based on the training set (70%) only. Now, using our final trained model stored in the best_model (et), we will predict the hold-out sample and evaluate the metrics to see if they are materially different from the CV results.

```
predict_model(et)

#The MAPE on the test/hold-out set is 0.0243 compared to 0.0228 achieved
on et CV result (in section above).
#This is not a significant difference.
```

Step 7: Finalize model for Deployment

```
final_et = finalize_model(et)
print(final_et)
```

Step 8: Predict on Unseen Data

The label column is added onto the data_unseen set. The label is the predicted value using the final_et model.

```
[182]  1  unseen_prediction = predict_model(final_et, data=data_unseen, round=0)
       2
       3  unseen_prediction.head()
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Extra Trees Regressor	6585	299223751	17298	1	0	0

	CustomerID	recency	frequency	monetary	Cluster	PersonID	TerritoryID	Mod
17	11017	106	4	6434.310059	Cluster 0	2521.0	9	
19	11019	16	33	882.700012	Cluster 0	5132.0	6	
23	11023	14	6	122.239998	Cluster 0	4373.0	4	
26	11026	105	7	6575.790039	Cluster 0	2857.0	9	
61	11061	228	6	8129.020020	Cluster 2	18220.0	9	

5 rows × 23 columns

Step 9: Saving the model

```
save_model(final_et, 'Final et model 1Apr2023')
```

Step 10: Loading the saved model

```
To load a saved model at a future date in the same or an alternative
environment, we would use PyCaret's load_model function and then easily
apply the model on new unseen data for prediction.
```

```
use this code:
* saved_final_et = load_model('Final et model 1Apr2023')
```

Results and recommedations

Based on the regression analysis for predicting lifetime value per customer, the MAPE (Mean Absolute Percentage Error) on the test/hold-out set is 0.0243, which is only slightly higher than the MAPE of 0.0228 achieved on the et CV result. This suggests that the model has performed well on the hold-out data and can effectively predict the lifetime value per customer.

However, it is important to keep in mind that the MAPE is just one measure of model performance and should be interpreted in the context of the business problem and other relevant metrics. For example, it would be useful to compare the predicted lifetime values to actual lifetime values to assess the accuracy of the model in absolute terms. Additionally, the business should consider the cost and benefits of implementing the model and determine if the predicted lifetime value information is useful for decision-making purposes.

In conclusion, the regression analysis shows that the model is capable of predicting lifetime value per customer with a reasonable degree of accuracy.

Comparing the two clusters of interest (Cluster 0 and Cluster 2)

Data Preparation

To analyze demographic characteristics, RFM values and lifetime values, the rfm_demographics data frame was checked and exported as a CSV file for visualization in PowerBI.

```
1  rfm_demographics.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 18409 entries, 0 to 18408
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   CustomerID            18409 non-null  object
 1   recency               18409 non-null  int32
 2   frequency             18409 non-null  int32
 3   monetary              18409 non-null  float32
 4   Cluster               18409 non-null  object
 5   lifetime_value        18409 non-null  float64
 6   PersonID              18409 non-null  float64
 7   TerritoryID           18409 non-null  int64
 8   AccountNumber         18409 non-null  object
 9   rowguid               18409 non-null  object
 10  ModifiedDate          18409 non-null  datetime64[ns]
 11  BusinessEntityID      18409 non-null  int64
 12  TotalPurchaseYTD      18409 non-null  float64
 13  DateFirstPurchase     18409 non-null  datetime64[ns]
 14  BirthDate             18409 non-null  datetime64[ns]
 15  MaritalStatus         18409 non-null  object
 16  YearlyIncome          18409 non-null  object
 17  Gender                18409 non-null  object
 18  TotalChildren         18409 non-null  float64
 19  NumberChildrenAtHome  18409 non-null  float64
 20  Education             18409 non-null  object
 21  Occupation            18409 non-null  object
 22  HomeOwnerFlag         18409 non-null  float64
 23  NumberCarsOwned       18409 non-null  float64
dtypes: datetime64[ns](3), float32(1), float64(7), int32(2), int64(2), object(9)
memory usage: 3.3+ MB
```

```
# to export rfm_demographics dataframe as CSV file
rfm_demographics.to_csv('rfm_demographics.csv', index=False)
```

Demographic characteristics

Who are our customers in Cluster 0 and cluster 2?

A data analysis was performed to study the overall characteristics of the customers in cluster 0 and cluster 2 to better understand the customers. In order to select the right cluster to focus on. By exploring the Figure 1, the following key statements can be drawn:

- There are more 50-75k yearly income in cluster 0 compared to cluster 2.
- There are slightly more married customers in cluster 0 compared to cluster 2.
- Customer´s gender distribution is almost the same in both clusters.
- The top occupation category is the “Professional” for both clusters.
- There are more Bachelors education in cluster 0 compared to cluster 2.
- There are more customers owning 2 cars in cluster 0 compared to cluster 2.
- There are more customers who do not have children in both clusters.

Number of registered customers
Cluster 0 : 10,398
Cluster 2 : 7,920

Figure 1. Customer distribution per demographic characteristics.

RFM values and lifetime value of each cluster

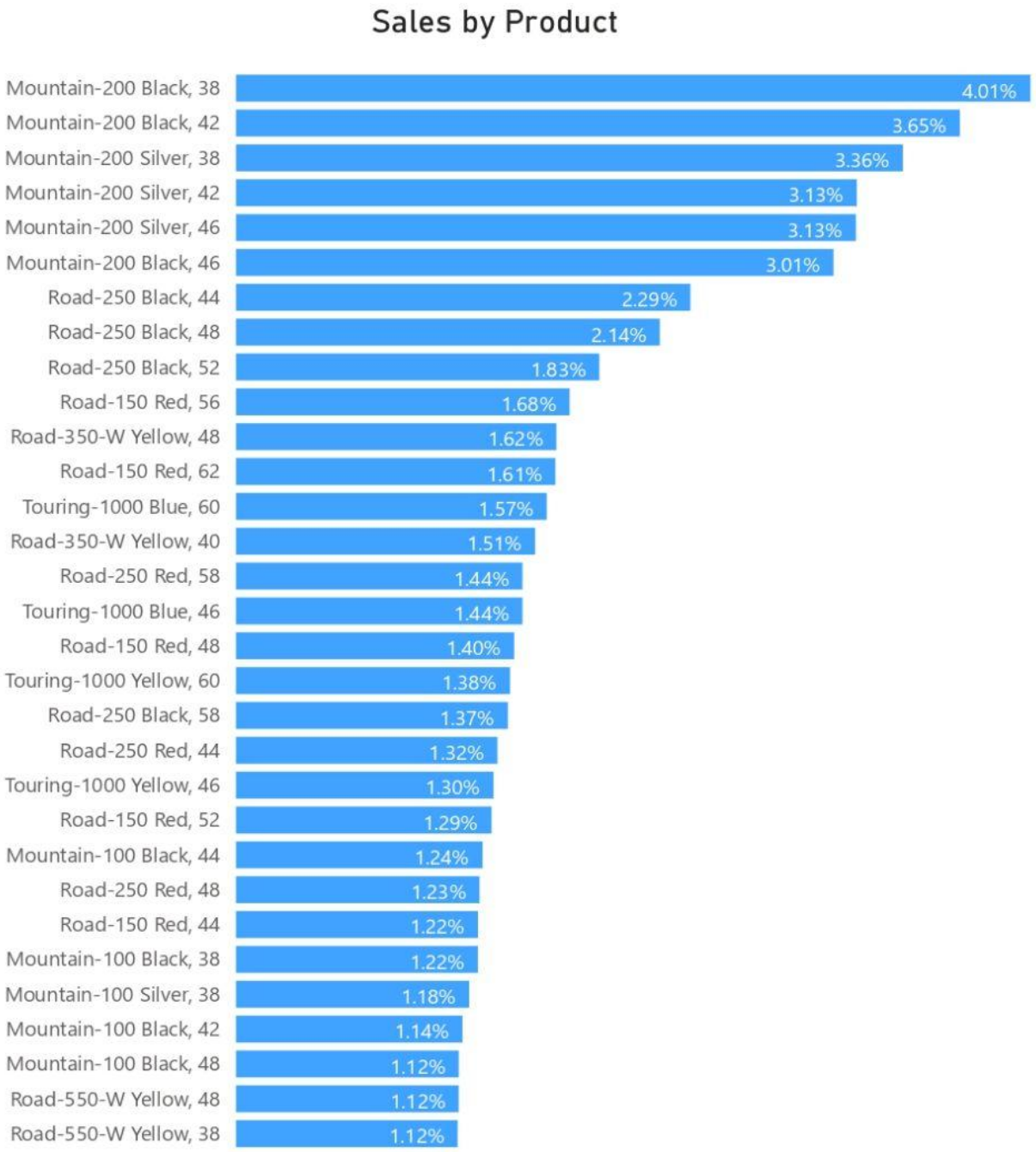
Based on the table below, the following statements are drawn:

- Customers in Cluster 0 have a higher mean for monetary value compared to customers in Cluster 2.
- Customers in Cluster 0 have a higher mean for frequency value compared to customers in Cluster 2.
- Customers in Cluster 0 have a higher mean for total purchase YTD and frequency compared to customers in Cluster 2.
- Customers in Cluster 0 have a higher mean for lifetime value compared to customers in Cluster 2.

Cluster	Average of monetary	Average of recency	Average of frequency	Average of TotalPurchaseYTD	Average of lifetime_value
Cluster 2	1,478.12	278.89	2.91	233.98	400,924.53
Cluster 0	1,633.33	95.78	3.63	475.12	452,026.22

Figure 2. Customers RFM values and lifetime value

A brief analysis by product name



Conclusion

Based on the data analysis, it can be concluded that customers in cluster 0 have a higher yearly income, more education, and a higher car ownership rate compared to customers in cluster 2. However, both clusters have a similar gender distribution and occupation category. Furthermore, the majority of customers in both clusters do not have children.

In terms of customer value, customers in cluster 0 have a higher mean for monetary value, frequency, total purchase YTD, and lifetime value compared to customers in cluster 2. Therefore, the company could consider focusing on cluster 0 as a priority since these customers are more valuable and generate more revenue for the company.

For the company's marketing strategies, they could tailor their offerings and promotions to better suit the demographic characteristics of each cluster. For example, for cluster 0, the company could offer more premium products or services, while for cluster 2, they could focus on affordability and value. Additionally, the company could use targeted advertising on social media platforms or email campaigns to reach out to each cluster more effectively.

However, if the company is constrained by budget or resources, it would be recommended to focus on cluster 0 first since these customers generate more revenue and have a higher lifetime value. Once the company has successfully targeted and acquired more customers in cluster 0, they can then shift their focus to cluster 2.