Business Data Understanding

1. Introduction

1.1 Problem statement

The business problem is to provide the Marketing Director an analysis into the Customer Lifetime value across segments to identify key segments.

The core segmentation that has been used is the industry of the customer which are;

- Food, drink and hospitality
- Retail
- Healthcare, beauty and fitness
- Professional services
- Home and repair
- Leisure and entertainment
- Online
- Travel and tourism
- Personal services
- Transportation

1.2 Methodology

For each segment, RFM segmentation has been used to carry out customer segmentation throughout the project because we can score the customers using

- Recency – Last time a customer made a purchase.
- Frequency – How often a customer makes a purchase.
- Monetary Value: Revenue – Expenses (500).

This gives us a clear picture of a customer behaves and can segment them into micro groups that can be efficiently targeted to improve retention. Particularly

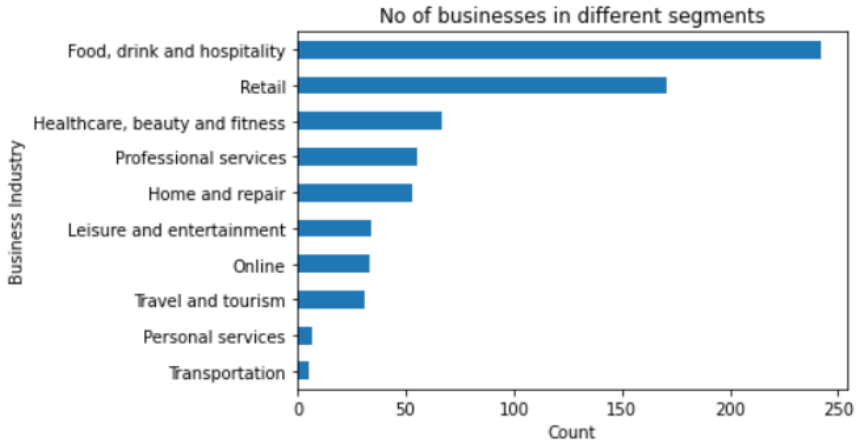
- Justify the Cost of Acquisition: Helps in determining the acceptable cost of acquisition and where to put more marketing effort.
- Potential customers: View the value of current and future customers.
- Customer Relationships: Helps to build stronger and efficient relations with customers.
- Helps to foster brand loyalty.

A high-level summary of the dataset is as follows.

Total Customers	Total Transactions	Total Transaction value
556	37,262	10,708,087

2. Customer Segments

The main segments from the data are



The number of businesses, count per segment, and Customer Lifetime Values is summarized as below.

Segment	No of Transactions	LTVCluster	Count (no. of businesses)	Mean score	Min score	Max score
Food, drink and hospitality	31919	Low value	156	44	0	251
		Mid value	24	486	276	811
		High Value	16	1,193	902	1,510
Retail	2508	Low value	112	31	0	125
		Mid value	12	234	139	367
		High Value	2	726	718	735
Healthcare, beauty and fitness	567	Low value	42	15	0	54
		Mid value	10	100	61	140
		High Value	4	235	200	292
Professional services	342	Low value	22	3	0	13
		Mid value	8	27	16	45
		High Value	4	74	58	93
Home and repair	529	Low value	26	20	0	68
		Mid value	8	144	91	268
		High Value	2	522	406	638
Leisure and entertainment	421	Low value	26	8	0	66
		Mid value	2	146	105	186
		High Value	1	320	320	320
Online	227	Low value	19	8	0	35
		Mid value	7	65	47	107
		High Value	1	208	208	208
Travel and tourism	499	Low value	10	15	0	37
		Mid value	9	69	44	92
		High Value	4	161	124	197
Personal services	149	Low value	1	10	10	10
		Mid value	2	15	15	15
		High Value	1	77	77	77
Transportation	101	NaN	NaN	NaN	NaN	NaN
Total	**37262**		**531**			

3.0 Recommendations

Majority of our customers are in the Low value segment. This means low recency, low frequency, and low monetary value. The business should strive to change this by pushing solutions such as

- Remove the charges for refunds, this may be discouraging them from returning.
- Create a rewards program to encourage repeat purchases.
- Revamp onboarding process for new customers, i.e. welcome email, free points upon signing up.

For the customers in Mid value, the business should

- Send targeted email campaigns with freebies and
- Use upsells to increase the transaction value.
- Collect more data such as age, gender to help understand and anticipate customer needs.

The high value segment contains our best customers, and should be kept under all circumstances.

- Make it super easy for these customers to reach customer support. They will feel their loyalty has a value.
- Send frequent emails, birthday messages, gifts to make them feel valued.
- Offer selected discounts, this will make them spend even more.

4. Metrics calculation

Pandas, Numpy was used to get the metrics below.

4.1 Average week processing times

Assumptions-

1. Report asked for a weekly average.
2. For all records with no filters.
3. Only positive time considered.
4. Blank values filled with median values.
5. Time is in milliseconds.
6. The lower the time the better.

```
transaction_df1 = transaction_df[transaction_df['processing_time'] > 0]

transaction_df['processing_time'].fillna(transaction_df['processing_time'].median(), inplace=True)

joined_df = pd.merge(transaction_df, business_df, how='inner', on = 'business_uuid')

joined_df['created'] = pd.to_datetime(joined_df['created'])

joined_df = joined_df[joined_df['sub_group'] == 'SEGMENT']

joined_df["created"] = joined_df["created"].dt.isocalendar().week

week_df = joined_df.groupby("created")["processing_time"].mean()

week_df
```

	AVG processing_time									
Week No.	Food, drink and hospitality	Retail	Healthcare, beauty and fitness	Professional services	Home and repair	Leisure and entertainment	Online	Travel	Personal services	Transportation
1	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-
3	15.10	-	-	15.10		-	-	-	-	-
4	15.10	-	-	-	-	-	-	-	-	-
5	16.62	-	-	-	-	26.32	26.46	-	-	-
6	19.02	-	-	-	-	-	-	-	-	-
7	20.30	-	-	17.83	-	15.10		-	19.30	-
8	14.23	16.38		16.29	-	19.63	11.53	-	14.75	-
9	15.10	15.18	15.10	13.86	-	17.82	17.02	-	17.05	-
10	10.15	15.50	15.10		-	13.21	2.42	-	11.58	-
11	64.74	11.87	15.10	7.44	-	27.68		-	29.87	-
12	12.33	17.82	15.10	22.36	-	-	-	-	-	-
13	12.96	13.05	12.88	4.59	-	20.24	25.02	-	18.41	-
14	16.14	0.00	17.24		-	-	-	-	31.03	11.86
15	13.71	14.31	13.28	15.10	-	5.15		-	13.97	38.15
16	16.24	15.78	13.38	19.07	-	-	10.51	-	13.18	56.27
17	14.85	13.39	10.27	15.26	-	12.71	14.13	-	20.07	35.91
18	14.64	21.30	13.75	14.41	-	-	-	-	9.17	15.33
19	15.65	21.05	19.76	12.05	-	9.10	13.77	-	45.10	22.99
20	16.19	13.93	17.41	12.01	-	10.24	4.54	-	30.95	16.67
21	53.91	15.72	30.75	6.38	5.20	-	15.43	-	18.42	16.11
22	16.65	19.31	16.05	15.10	17.07	13.42	18.53	-	15.08	14.62
23	33.38	18.57	15.28	20.52	-	17.34	-	16.575	24.02	8.57
24	15.89	14.21	17.86	16.34	9.69	-	-	25.548	29.61	15.47
25	33.29	13.99	11.09	16.02	-	20.91	15.10	32.108	13.91	12.32
26	169.41	16.79	12.84	11.07	14.51	16.58		16.398	28.53	17.45
27	29.46	18.55	42.00	22.46	16.03	-	16.01	20.473	16.39	16.67
28	22.80	17.55	21.64	15.45	29.71	-	-	25.37	11.73	18.95
29	17.34	19.62	25.09	85.54	14.91	6.95	228.44	21.868	18.15	15.32
30	17.12	18.02	30.22	19.67	22.12	13.97	17.57	24.027	17.89	6.49
31	17.15	19.52	14.84	14.43	24.87	8.34	-	14.938	23.30	7.18
32	56.64	17.75	22.03	17.11	20.35	21.70	11.16	16.696	16.27	7.52
33	22.78	19.49	15.30	21.67	16.86	12.28	-	16.147	22.02	15.19
34	17.42	20.52	16.92	11.82	23.97	15.10	16.77	39.668	27.15	21.61
35	17.93	22.91	19.24	8.49	18.46	19.66	9.48	14.844	26.90	15.10
36	18.19	17.89	27.88	14.31	19.47	17.05	-	24.45	-	13.95
37	18.36	18.54	21.27	13.13	12.65	34.19	21.40	19.264	25.19	10.44
38	18.90	17.60	34.55	18.97	19.35	24.16	-	18.43	17.91	16.19
39	23.33	17.52	15.62	14.90	18.62	23.66	3279.79	17.738	36.00	-
40	20.85	21.67	14.26	16.73	22.65	29.17	-	19.071	25.69	-
41	22.52	16.53	68.49	14.65	14.95	20.30	7.52	16.987	34.18	-
42	22.93	19.32	13.91	15.10	32.86	70.44	13.17	24.074	23.29	-
43	31.88	21.52	20.49	18.48	51.85	21.56	15.36	21.492	19.40	-
44	20.01	48.02	19.04	23.13	23.52	19.66	130.46	19.663	21.69	-
45	66.39	22.42	19.25	24.41	202.32	22.16	11.52	19.801	18.50	-

	AVG processing_time									
Week No.	**Food, drink and hospitality**	**Retail**	**Healthcare, beauty and fitness**	**Professional services**	**Home and repair**	**Leisure and entertainment**	**Online**	**Travel**	**Personal services**	**Transportation**
46	22.48	17.14	18.68	14.89	17.21	19.81	28.23	18.692	6.03	-
47	19.47	20.05	16.73	14.07	20.62	32.17	19.69	21.543	13.84	-
48	19.11	17.83	19.86	25.27	18.36	23.22	19.80	25.209	16.00	-
49	19.75	17.91	431.88	12.06	28.28	27.19	17.59	27.499	27.00	-
50	18.47	17.94	15.45	24.97	24.10	23.25	17.43	17.529	29.92	-
51	48.29	19.77	-	18.34	19.47	24.50	14.50	22.537	25.61	-
52	22.75	20.42	-	-	70.95	11.77	14.19	18.017	24.11	-
53	42.94	16.53	-	16.16	34.13	16.39	8.81	22.515	19.56	-
AVG	26.29	18.06	29.69	17.49	28.55	20.10	124.04	21.26	21.50	17.85

4.2 Transaction approval rate

assumptions – only transactions with positive time are considered.

```
approved_count = (joined_df['transaction_state'] == 'approved').sum().sum()
count_row = joined_df['transaction_uuid'].shape[0]
approval_rate = approved_count/count_row*100
approval_rate
```

Transaction Approval Rate			
Segment	Total Transactions	Approved Transactions	Approval rate
Food, drink and hospitality	31919	26200	82.1%
Retail	2508	2062	82.2%
Healthcare, beauty and fitness	567	450	79.4%
Professional services	342	230	67.3%
Home and repair	529	430	81.3%
Leisure and entertainment	421	315	74.8%
Online	227	163	71.8%
Travel and tourism	499	418	83.8%
Personal services	149	127	85.2%
Transportation	101	46	45.5%

4.3 Credit card transaction rate

Assumptions

1. only transactions with positive time are considered.

2. Only credit card transaction type is considered.

```
approved_count = (joined_df['transaction_type'] == 'credit_card').sum().sum()

count_row = joined_df['transaction_uuid'].shape[0]

approval_rate = approved_count/count_row*100

approval_rate
```

Credit Card Transaction Rate			
Segment	Total Transactions	Credit Card Transactions	Approval rate
Food, drink and hospitality	31335	31919	98.2%
Retail	2508	2290	91.3%
Healthcare, beauty and fitness	567	520	91.7%
Professional services	342	246	71.9%
Home and repair	529	495	93.6%
Leisure and entertainment	421	410	97.4%
Online	227	195	85.9%
Travel and tourism	499	325	65.1%
Personal services	149	142	95.3%
Transportation	101	100	99.0%

4.4 Average Transactions per week

Assumptions made

1. Only transactions with positive time considered.
2. A week defined as having 7 days.
3. Majority of the transactions are skewed towards the end of the year.

```
joined_df = joined_df[joined_df['sub_group'] == 'SEGMENT']

joined_df["created_day"] = joined_df["created"].dt.day

avg_tr_day_df = joined_df.groupby(['created','created_day'])['transaction_uuid'].count().reset_index(name="count_daily")

weekly_sum_df = (avg_tr_day_df.groupby([avg_tr_day_df["created"].dt.isocalendar().week], sort=False)
  .count_daily.agg([('Sum','sum')])
  .add_prefix('daily_count')
  .reset_index()
)

weekly_avg = weekly_sum_df['daily_countSum']

weekly_avg.round(0)
```

	Average No. of transactions per week										
Week No.	Food, drink and hospitality	Retail	Healthcare, beauty and fitness	Professional services	Home and repair	Leisure and enta	Online	Tourism & Travel	Personal services	Transpo-rtation	TOTAL
1	-	-	-	-	-	-	-	-	-	-	0
2	-	-	-	-	-	-	-	-	-	-	0
3	7	-	-	-	-	-	-	-	-	-	7
4	6	-	-	-	-	-	-	-	-	-	6
5	5	-	-	-	-	-	-	-	-	-	5
6	18	-	-	-	-	-	-	-	-	-	18
7	1	-	-	-	-	-	-	-	-	-	1
8	2	-	-	-	-	-	-	-	-	-	2
9	1	5	-	3	-	-	-	-	2	-	11
10	2	15	-	6	-	-	-	-	4	-	27
11	1	6	-	2	-	-	-	-	2	-	11
12	1	5	2	2	-	-	-	-	3	-	13
13	10	8	2	3	-	-	-	-	1	-	24
14	39	8	1	14	-	5	-	-	2	-	69
15	38	11	1	1	-	1	-	-	4	-	56
16	65	8	9	1	-	4	-	-	8	-	95
17	136	14	3	3	-	9	-	-	3	-	168
18	172	17	4	4	-	3	-	-	3	-	203
19	178	20	6	2	-	4	-	-	3	-	213
20	226	17	3	8	-	8	-	-	4	-	266
21	239	33	1	4	-	1	1	-	3	-	282
22	259	41	9	2	-	6	1	-	1	-	319
23	318	43	4	3	2	5	2	2	4	-	383
24	388	30	13	6	1	4	1	1	5	-	449
25	410	48	4	5	3	6	1	3	5	-	485
26	447	34	3	7	2	6	1	1	2	-	503
27	511	47	5	8	8	15	2	2	1	-	599
28	462	49	2	6	4	4	2	3	10	-	542
29	526	34	8	7	7	1	1	11	2	1	598
30	556	64	10	7	11	8	1	6	2	6	671
31	677	57	8	6	5	5	2	18	3	2	783
32	778	71	13	13	6	10	3	19	6	2	921
33	677	44	5	7	6	8	1	18	3	4	773
34	699	30	6	5	4	1	1	15	3	2	766
35	797	44	10	12	10	6	9	16	2	4	910
36	848	36	8	18	5	7	2	11	2	4	941
37	948	46	7	19	5	4	6	9	2	3	1049
38	704	36	14	11	12	4	5	9	4	4	803
39	1016	61	14	18	8	23	9	15	2	2	1168
40	1150	58	21	9	5	6	16	13	1	2	1281
41	1273	51	18	7	11	12	4	15	5	4	1400
42	1215	46	17	7	9	1	1	23	3	4	1326
43	1283	54	19	1	13	6	13	24	4	11	1428
44	1344	131	16	11	21	23	7	20	1	3	1577

	Average No. of transactions per week										
Week No.	Food, drink and hospitality	Retail	Healthcare, beauty and fitness	Professional services	Home and repair	Leisure and enta	Online	Tourism & Travel	Personal services	Transpo-rtation	TOTAL
45	1370	115	21	13	20	15	11	15	5	4	**1589**
46	1368	89	23	16	34	12	8	14	1	7	**1572**
47	1462	96	27	11	27	9	6	20	3	2	**1663**
48	1672	179	34	19	47	88	20	25	7	2	**2093**
49	1747	163	52	12	24	11	25	30	1	1	**2066**
50	1708	154	54	7	48	26	23	28	4	1	**2053**
51	1931	185	58	10	87	33	18	36	6	18	**2382**
52	1445	164	26	4	61	7	22	52	2	6	**1789**
53	783	41	6	2	23	14	2	25	5	2	**903**
											37262