

## ▼ ANALYSIS ON AMAZON : FACING LOW CUSTOMER SATISFACTION IN SINGAPORE

### **Objective**

***To analyse and gain insights into the performance and trends of the Amazon in Singapore. The project aims to provide actionable recommendations to improve business strategies and enhance overall performance.***

### ▼ Attribute information

uid - Unique ID

company\_v - Companies - Open-end

poverq - Overall Product Quality

soverq - Overall Service Quality

pq - Price given quality

satis - Customer Satisfaction

repur - Likelihood to Repurchase

recomm - Likelihood to recommend

Q19 Recommended - ecommerce site to family and friends in the last 3 months

VN\_1009\_Q20A - Overall experiences SATISFACTION with (INSERT NAME)

VN\_1009\_TP01 - Variety of products that interests me

VN\_1009\_TP02 - Variety of products that meet my needs

VN\_1009\_TP03 - Ease of navigating the website or app

VN\_1009\_TP04 - Ease of finding the products you need

VN\_1009\_TP05 - Availability of products

VN\_1009\_TP06 - Attractiveness of promotions and discounts offered

VN\_1009\_TP07 - Sufficiency of Product information

VN\_1009\_TP08 - Ease of comparing products

VN\_1009\_TP09 - Ease of indicating special requests  
VN\_1009\_TP10 - Ease of managing your shopping cart  
VN\_1009\_TP11 - Check-out and payment process  
VN\_1009\_TP12 - Security of website  
VN\_1009\_TP13 - Clarity and usefulness of information on your delivery methods and fees  
VN\_1009\_TP14 - Range of delivery options  
VN\_1009\_TP15 - Ease of tracking your order  
VN\_1009\_TP16 - Time taken to receive the product  
VN\_1009\_TP17 - Products you received were as described on the website  
VN\_1009\_TP18 - Availability of feedback channels  
VN\_1009\_TP19 - Return and exchange policies  
VN\_1009\_TP20 - Method Used most frequently to shop at (INSERT NAME)  
VN\_1009\_TP21 - Method of payment do you prefer most for shopping online at (INSERT NAME)  
VN\_1009\_TP21\_6specify - Method of payment do you prefer most for shopping online at (INSERT NAME) Other specify  
VN\_1009\_TP24\_1 - Read the reviews during shopping experience  
VN\_1009\_TP24\_2 - Interacted directly with the seller during shopping experience  
VN\_1009\_TP22 - Satisfaction with the product reviews  
VN\_1009\_TP23 - Satisfaction with the channels available to communicate with the seller(s)  
Q9C\_P - No. of times purchased in the last 6 months  
Q9D - Average amount spent per visit last 3 months  
VN\_1009\_TP25A - General Shopping Behavior: Usually make most purchases on physical store or an online store  
age - Age  
race - Race of the respondent  
work - Employment status of the respondent  
pincome - Monthly Personal Income  
income - Monthly Household Income  
educat - Education Qualification  
childsupp - No. of children dependent for financial support  
marital - Marital Status  
gender - Gender

house - Housetype according to respondent

DOI - Date of Interview

```
import pandas as pd
df = pd.read_excel(r'/content/Amazon_Facing Low Customer Satisfaction in Singapore_test111111.xlsx')
df
```

	uid	company_v	poverq	soverq	pq	satis	repur	recomm	Q19	VN_1009_Q20A	...	race	work	pincome	income	educat
<b>0</b>	5	ZALORA	9	7	6	7	6	5	0	8	...	1	1	2	4	8
<b>1</b>	11	FAVE	7	8	8	7	7	6	0	8	...	1	1	2	4	7
<b>2</b>	15	FAVE	6	7	7	6	6	6	0	7	...	1	1	4	5	8
<b>3</b>	19	AMAZON	8	8	7	8	8	6	0	8	...	1	1	3	5	8
<b>4</b>	20	QOO10	7	6	8	8	6	5	0	7	...	1	1	4	5	8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
<b>1595</b>	6066	FAVE	7	7	7	8	6	7	0	8	...	1	3		7	7
<b>1596</b>	6068	EBAY	7	9	9	8	9	6	0	7	...	1	1	7	8	7
<b>1597</b>	6070	ZALORA	6	6	8	7	9	9	0	8	...	1	2	3	8	7
<b>1598</b>	6079	TAOBAO/TMALL	7	8	8	7	7	8	0	8	...	1	1	4	9	8
<b>1599</b>	6087	AMAZON	6	6	6	8	9	8	1	9	...	1	1	3	7	8

1600 rows × 50 columns



```
df.head()
```

↗

	uid	company_v	poverq	soverq	pq	satis	repur	recomm	Q19	VN_1009_Q20A	...	race	work	pincome	income	educat	childsuj
0	5	ZALORA	9	7	6	7	6	5	0	8	...	1	1	2	4	8	
1	11	FAVE	7	8	8	7	7	6	0	8	...	1	1	2	4	7	
2	15	FAVE	6	7	7	6	6	6	0	7	...	1	1	4	5	8	
3	19	AMAZON	8	8	7	8	8	6	0	8	...	1	1	3	5	8	
4	20	QOO10	7	6	8	8	6	5	0	7	...	1	1	4	5	8	

5 rows × 50 columns



df.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1600 entries, 0 to 1599
Data columns (total 50 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   uid                   1600 non-null   int64
 1   company_v            1600 non-null   object
 2   poverq               1600 non-null   int64
 3   soverq               1600 non-null   int64
 4   pq                   1600 non-null   int64
 5   satis                1600 non-null   int64
 6   repur                1600 non-null   int64
 7   recomm               1600 non-null   int64
 8   Q19                  1600 non-null   int64
 9   VN_1009_Q20A         1600 non-null   int64
10   VN_1009_TP01         1600 non-null   int64
11   VN_1009_TP02         1600 non-null   int64
12   VN_1009_TP03         1600 non-null   int64
13   VN_1009_TP04         1600 non-null   int64
14   VN_1009_TP05         1600 non-null   int64

```

15	VN_1009_TP06	1600	non-null	int64
16	VN_1009_TP07	1600	non-null	int64
17	VN_1009_TP08	1600	non-null	int64
18	VN_1009_TP09	1600	non-null	int64
19	VN_1009_TP10	1600	non-null	int64
20	VN_1009_TP11	1600	non-null	int64
21	VN_1009_TP12	1600	non-null	int64
22	VN_1009_TP13	1600	non-null	int64
23	VN_1009_TP14	1600	non-null	int64
24	VN_1009_TP15	1600	non-null	int64
25	VN_1009_TP16	1600	non-null	int64
26	VN_1009_TP17	1600	non-null	int64
27	VN_1009_TP18	1600	non-null	int64
28	VN_1009_TP19	1600	non-null	int64
29	VN_1009_TP20	1600	non-null	int64
30	VN_1009_TP21	1600	non-null	int64
31	VN_1009_TP21_6specify	1600	non-null	object
32	VN_1009_TP24_1	1600	non-null	int64
33	VN_1009_TP24_2	1600	non-null	int64
34	VN_1009_TP22	1600	non-null	object
35	VN_1009_TP23	1600	non-null	object
36	Q9C_P	1600	non-null	int64
37	Q9D	1600	non-null	int64
38	VN_1009_TP25A	1600	non-null	int64
39	age	1600	non-null	int64
40	race	1600	non-null	int64
41	work	1600	non-null	int64
42	pincome	1600	non-null	object
43	income	1600	non-null	int64
44	educat	1600	non-null	int64
45	childsupp	1600	non-null	int64
46	marital	1600	non-null	int64
47	gender	1600	non-null	int64
48	house	1600	non-null	int64
49	DOI	1600	non-null	object

dtypes: int64(44), object(6)

memory usage: 625.1+ KB

df.describe()

	uid	poverq	soverq	pq	satis	repur	recomm	Q19	VN_1009_Q20A	VN_
<b>count</b>	1600.000000	1600.000000	1600.000000	1600.000000	1600.000000	1600.000000	1600.000000	1600.000000	1600.000000	1600.000000
<b>mean</b>	2948.575000	7.573750	7.548750	7.573125	7.575625	7.44625	7.290625	0.248125	7.669375	7.669375
<b>std</b>	1717.270229	1.362248	1.314802	1.363200	1.192233	1.27451	1.357779	0.432060	1.243897	1.243897
<b>min</b>	5.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	0.000000	2.000000	2.000000
<b>25%</b>	1478.500000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	0.000000	7.000000	7.000000
<b>50%</b>	2877.000000	8.000000	8.000000	8.000000	8.000000	7.000000	7.000000	0.000000	8.000000	8.000000
<b>75%</b>	4395.500000	8.000000	8.000000	8.000000	8.000000	8.000000	8.000000	0.000000	8.000000	8.000000

**1. Companies often benchmark themselves on various key performance indicators ratings. How does Amazon.com perform on these key metrics as compared to its competitors?**

- Customer Satisfaction
- Willingness to recommend
- Average customer spend
- Frequency of visit

```
# Separating Amazon data
amazon_data = df[df['company_v'] == 'AMAZON']

# Separating competitors data
competitors_data = df[df['company_v'] != 'AMAZON']

# To calculate average customer_satisfaction for Amazon
amazon_customer_satisfaction = amazon_data['satis'].mean()
# To calculate average customer_satisfaction for competitors
competitors_customer_satisfaction = competitors_data['satis'].mean()

print("Amazon Customer Satisfaction:", amazon_customer_satisfaction)
print("Competitors Customer Satisfaction:", competitors_customer_satisfaction)
```

```
# To calculate average willingness to recommend for Amazon
amazon_willingness_to_recommend = amazon_data['recomm'].mean()
# To calculate average Willingness to Recommend for competitors
competitors_willingness_to_recommend = competitors_data['recomm'].mean()

print("Willingness to recommend Amazon:", amazon_willingness_to_recommend)
print("Willingness to recommend Competitors:", competitors_willingness_to_recommend)

# To calculate average Average Customer Spend for Amazon
amazon_average_customer_spend = amazon_data['Q9D'].mean()
# To calculate average Average Customer Spend for competitors
competitors_average_customer_spend = competitors_data['Q9D'].mean()

print("Average customer spend for Amazon:", amazon_average_customer_spend)
print("Average customer spend for Competitors:", competitors_average_customer_spend)

# To calculate average Frequency of Visit for Amazon
amazon_frequency_of_visit = amazon_data['Q9C_P'].mean()
# To calculate average Frequency of Visit for competitors
competitors_frequency_of_visit = competitors_data['Q9C_P'].mean()

print("frequency of visit for Amazon:", amazon_frequency_of_visit)
print("frequency of visit for Competitors:", competitors_frequency_of_visit)
```

```
Amazon Customer Satisfaction: 7.48
Competitors Customer Satisfaction: 7.589285714285714
Willingness to recommend Amazon: 7.17
Willingness to recommend Competitors: 7.307857142857143
Average customer spend for Amazon: 154.45
Average customer spend for Competitors: 168.36428571428573
frequency of visit for Amazon: 2.125
frequency of visit for Competitors: 2.2642857142857142
```

```
# Comparing Amazon's performance with competitors
if amazon_customer_satisfaction > competitors_customer_satisfaction:
    print("Amazon has a higher Customer Satisfaction than competitors.")
elif amazon_customer_satisfaction < competitors_customer_satisfaction:
```

```
print("Amazon has a lower Customer Satisfaction than competitors.")
else:
    print("Amazon's Customer Satisfaction is on par with competitors.")

# Comparing Amazon's performance with competitors
if amazon_willingness_to_recommend > competitors_willingness_to_recommend:
    print("Amazon has a higher Willingness to Recommend than competitors.")
elif amazon_willingness_to_recommend < competitors_willingness_to_recommend:
    print("Amazon has a lower Willingness to Recommend than competitors.")
else:
    print("Amazon's Willingness to Recommend is on par with competitors.")
```

```
# Comparing Amazon's performance with competitors
if amazon_average_customer_spend > competitors_average_customer_spend:
    print("Amazon has a higher Average Customer Spend than competitors.")
elif amazon_average_customer_spend < competitors_average_customer_spend:
    print("Amazon has a lower Average Customer Spend than competitors.")
else:
    print("Amazon's Average Customer Spend is on par with competitors.")
```

```
# Comparing Amazon's performance with competitors
if amazon_frequency_of_visit > competitors_frequency_of_visit:
    print("Amazon has a higher Frequency of Visit than competitors.")
elif amazon_frequency_of_visit < competitors_frequency_of_visit:
    print("Amazon has a lower Frequency of Visit than competitors.")
else:
    print("Amazon's Frequency of Visit is on par with competitors.")
```

```
Amazon has a lower Customer Satisfaction than competitors.
Amazon has a lower Willingness to Recommend than competitors.
Amazon has a lower Average Customer Spend than competitors.
Amazon has a lower Frequency of Visit than competitors.
```

```
import matplotlib.pyplot as plt
```

```
# Creating a bar plot
metrics1 = ('Amazon Customer Satisfaction', 'Willingness to recommend Amazon',
```



```
        'Average customer spend for Amazon', 'frequency of visit for Amazon')
values1 = (amazon_customer_satisfaction, amazon_willingness_to_recommend,
          amazon_average_customer_spend, amazon_frequency_of_visit)
```

```
plt.figure(figsize=(8, 6))
plt.bar(metrics1, values1, color=('BLUE','BROWN','RED','PINK'))
plt.title('Amazon Performance Metrics')
plt.xlabel('Metrics')
plt.ylabel('Average Value')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

## Amazon Performance Metrics

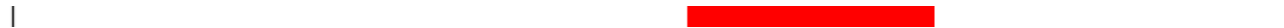


▼ 2.



**From the data, what are the reasons for Amazon's performance ratings?**

- Which areas did Amazon perform well in?
- Which areas did Amazon perform poorly in?



```
# To calculate the correlation matrix to find relationships between variables
correlation_matrix = df.corr()
```

```
# Listing the performance-related variables
```

```
performance_variables = [
    'poverq',
    'soverq',
    'pq',
    'satis',
    'repur',
    'recomm',
    'VN_1009_TP05',
    'VN_1009_TP07',
    'VN_1009_TP19'
]
```

```
# To calculate the average performance ratings for Amazon
amazon_performance_ratings = df[df['company_v'] == 'AMAZON'][performance_variables].mean()
```

```
# Printing Amazon's performance ratings for each variable
print("Amazon's Performance Ratings:")
for variable, rating in amazon_performance_ratings.items():
    print(f"{variable}: {rating:.2f}")
```

```
# Identifying areas where Amazon performed well
highly_performing_areas = amazon_performance_ratings[amazon_performance_ratings >= 8.0].index.tolist()

print("\nAreas where Amazon performed well:")
for area in highly_performing_areas:
    print(area)
```

```
# Identifying areas where Amazon performed poorly
poorly_performing_areas = amazon_performance_ratings[amazon_performance_ratings <= 8.0].index.tolist()

print("\nAreas where Amazon performed poorly:")
for area in poorly_performing_areas:
    print(area)
```

Amazon's Performance Ratings:

poverq: 7.49

soverq: 7.54

pq: 7.41

satis: 7.48

repur: 7.38

recomm: 7.17

VN\_1009\_TP05: 7.58

VN\_1009\_TP07: 7.59

VN\_1009\_TP19: 25.74

Areas where Amazon performed well:

VN\_1009\_TP19

Areas where Amazon performed poorly:

poverq

soverq

pq

satis

repur

recomm

VN\_1009\_TP05

VN\_1009\_TP07

```
<ipython-input-10-1a173a775611>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it w
correlation_matrix = df.corr()
```

### ▼ 3.

#### ***What should Amazon do to improve its customer satisfaction performance?***

```
# To calculate the average performance ratings for Amazon
amazon_performance_ratings = df[df['company_v'] == 'AMAZON'][performance_variables].mean()

# Defining a dictionary to store improvement strategies and their descriptions
improvement_strategies = {
    'poverq': 'Implement stricter quality control processes and ensure products meet or exceed customer expectations.',
    'soverq': 'Invest in customer service training, reduce response times, and focus on effective issue resolution.',
    'pq': 'Evaluate pricing strategies to ensure they align with the perceived value of products and services.',
}

# Printing Amazon's performance ratings for each variable
print("Amazon's Performance Ratings:")
for variable, rating in amazon_performance_ratings.items():
    print(f"{variable}: {rating:.2f}")

# Printing recommended improvement strategies based on areas of lower performance
print("\nRecommended Improvement Strategies:")
for area, strategy in improvement_strategies.items():
    if amazon_performance_ratings[area] <= 8.0:
        print(f"{area}: {strategy}")

Amazon's Performance Ratings:
poverq: 7.49
soverq: 7.54
pq: 7.41
satis: 7.48
repur: 7.38
```

```

recomm: 7.17
VN_1009_TP05: 7.58
VN_1009_TP07: 7.59
VN_1009_TP19: 25.74

```

Recommended Improvement Strategies:

poverq: Implement stricter quality control processes and ensure products meet or exceed customer expectations.

soverq: Invest in customer service training, reduce response times, and focus on effective issue resolution.

pq: Evaluate pricing strategies to ensure they align with the perceived value of products and services.

## ▼ 4.

***Compare the performance of Amazon with that of Qoo10?***

***a) How is Qoo10 performing relative to Amazon?***

***b) What should Amazon do to improve its performance?***

***c) What should Amazon do to do better compete with Qoo10?***

```

performance_variables = [
    'poverq',
    'soverq',
    'pq',
    'satis',
    'repur',
    'recomm',
    'VN_1009_TP05',
    'VN_1009_TP07',
    'VN_1009_TP19',
    'VN_1009_TP01',
    'VN_1009_TP02',
    'VN_1009_TP15',
    'VN_1009_TP16'
]

```

# Filtering the data for Qoo10

```
Q0010_data = df[df['company_v'] == 'Q0010']
```

```
# To calculate the average performance ratings for Amazon
amazon_performance_ratings = amazon_data[performance_variables].mean()

# To calculate the average performance ratings for Qoo10
Q0010_performance_ratings = Q0010_data[performance_variables].mean()

# Printing performance ratings for Amazon and Qoo10
print("Amazon's Performance Ratings:")
print(amazon_performance_ratings)

print("\nQ0010's Performance Ratings:")
print(Q0010_performance_ratings)

# Comparing Qoo10's performance relative to Amazon
performance_comparison = Q0010_performance_ratings - amazon_performance_ratings
print("\nPerformance Comparison (Q0010 relative to Amazon):")
print(performance_comparison)
```

Amazon's Performance Ratings:

poverq	7.490
soverq	7.545
pq	7.410
satis	7.480
repur	7.380
recomm	7.170
VN_1009_TP05	7.585
VN_1009_TP07	7.590
VN_1009_TP19	25.740
VN_1009_TP01	7.905
VN_1009_TP02	7.505
VN_1009_TP15	20.125
VN_1009_TP16	7.490

dtype: float64

Q0010's Performance Ratings:

poverq	7.610
soverq	7.505
pq	7.560
satis	7.580
repur	7.320
recomm	7.290

```

VN_1009_TP05      7.590
VN_1009_TP07      7.540
VN_1009_TP19     25.255
VN_1009_TP01      8.040
VN_1009_TP02      7.550
VN_1009_TP15     21.300
VN_1009_TP16      7.800
dtype: float64

```

Performance Comparison (Q0010 relative to Amazon):

```

poverq      0.120
soverq     -0.040
pq           0.150
satis        0.100
repur       -0.060
recomm       0.120
VN_1009_TP05  0.005
VN_1009_TP07 -0.050
VN_1009_TP19 -0.485
VN_1009_TP01  0.135
VN_1009_TP02  0.045
VN_1009_TP15  1.175
VN_1009_TP16  0.310
dtype: float64

```

## ▼ REPORT

The provided dataset is a subset of the survey data collected from Singaporeans and Permanent Residents via a randomised door-to-door methodology. Respondents were asked about their customer experience with the respective e-commerce website they had recently used in the last 3 months. Data were collected from January to April 2018 with a total of N = 1600 samples.

As shown in the analysis, it is clear that Amazon's performance is low in several areas and Hence, they're facing a low customer satisfaction .As done in Q.3, they can improve their customer satisfaction by considering those variables or more.

From the Q.4, Q0010 performs better in more areas than Amazon. Also, analysis in Q.2 says that amazon performs very well only in return and exchange policy among the considered attributes.

Hence, overall performance of Amazon is critical in the market of Singapore.

---

✓ 0s completed at 15:32

