PART B(Simmons data Analysis)

In [1]:
```
#Imports the Necessary libraries
import pandas as pd
from sklearn.linear_model import LogisticRegression
import numpy as np
print("Libraries imported successfully!")
```

```
Libraries imported successfull!
```

In [2]:
```
# Load the dataset
file_path = "C:\\Users\\n\\Downloads\\Simmons-data-raw.xls"
data = pd.read_excel(file_path)
```

In [3]:
```
data.head()
```

Out[3]:

	Simmons-data-raw	Unnamed: 1	Unnamed: 2	Unnamed: 3
0	Customer	Spending(000)	Card	Coupon-Usage-Indicator
1	1	2.291	1	0
2	2	3.215	1	0
3	3	2.135	1	0
4	4	3.924	0	0

In [4]:
```
data.tail()
```

Out[4]:

	Simmons-data-raw	Unnamed: 1	Unnamed: 2	Unnamed: 3
96	96	3.318	0	0
97	97	2.421	1	0
98	98	6.073	0	0
99	99	2.63	1	0
100	100	3.411	0	1

In [6]:
```
# Split the data into predictors (X1 and X2) and the target variable (Y)
data=pd.read_excel("C:\\Users\\n\\Downloads\\Simmons-data-raw(1).xls")
X = data[['Spending(000)', 'Card']]
Y = data['Coupon-Usage-Indicator']
```

In [8]:
```
# Build a logistic regression model
logistic_model = LogisticRegression()
logistic_model.fit(X, Y)
print('Build Successful!')
```

```
Build Successful!
```

Coefficients of the logistic regression model

In [12]:
```
# PartB-1: Coefficients of the logistic regression model
beta0 = logistic_model.intercept_[0]
beta1, beta2 = logistic_model.coef_[0]
```

In [14]:
```
# PartB-2: Probability of response for Jack and Jill
# Jack: X1 = 2, X2 = 1
# Jill: X1 = 4, X2 = 0
jack_data = np.array([[2, 1]])
jill_data = np.array([[4, 0]])
jack_probability = logistic_model.predict_proba(jack_data)[:, 1]
jill_probability = logistic_model.predict_proba(jill_data)[:, 1]
```

```
C:\Users\n\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X do
es not have valid feature names, but LogisticRegression was fitted with featu
re names
  warnings.warn(
C:\Users\n\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X do
es not have valid feature names, but LogisticRegression was fitted with featu
re names
  warnings.warn(
```

Choosing the cutoff probability

In [16]:
```
# PartB-3: Choosing the cutoff probability
# You can evaluate different cutoff probabilities using a confusion matrix
# and choose the one that best suits your business needs.

# For example, you can set a cutoff probability of 0.5, where probabilities >=
cutoff_probability = 0.5
```

RESULTS

In [17]:

```
# Print the results
print("LR coefficients Value")
print(f"BETA0 (or constant term): {beta0}")
print(f"BETA1 (coeff. For X1): {beta1}")
print(f"BETA2 (coeff. For X2): {beta2}")

print("Probability of Response")
print(f"Jack: {jack_probability[0]}")
print(f"Jill: {jill_probability[0]}")

if jack_probability > jill_probability:
    print("Jack is more likely to respond because he has a higher probability.
else:
    print("Jill is more likely to respond because she has a higher probability
```

```
LR coefficients Value
BETA0 (or constant term): -2.006720615442227
BETA1 (coeff. For X1): 0.32989442356674026
BETA2 (coeff. For X2): 0.9178862828888503
Probability of Response
Jack: 0.3943542838553842
Jill: 0.3346689457446288
Jack is more likely to respond because he has a higher probability.
```

In [ ]:

```
#THE END
```