

# DATA ANALYST

## Intenship Task 15

### DESCRIPTION

This task focuses on performing Customer Segmentation using RFM (Recency, Frequency, Monetary) analysis on an e-commerce dataset. The objective is to identify valuable customer groups based on purchasing behavior and generate meaningful business insights to support targeted marketing and strategic decision-making.

### PREPARED BY

Reema Safrin M  
(12-02-2026)

### MY WORK

In this task, I performed RFM analysis using the provided e-commerce dataset. I first cleaned and prepared the data by converting date columns and validating customer-level records. Then, I calculated Recency, Frequency, and Monetary values for each customer using groupby operations. Using quantile-based scoring, I assigned RFM scores and combined them to classify customers into segments such as Champions, Loyal Customers, At Risk, and Others. I also visualized the segment distribution and exported the final results as a CSV file. This project helped me understand customer behavior analysis and how data-driven segmentation supports strategic marketing decisions.

#### DATASET USED

[ecommerce\\_data](#)

#### FINAL DATASET

[rfm\\_segments](#)

# Customer Segmentation (RFM Analysis)

## Step-by-Step Process Using ecommerce\_data.csv

### 1. Import Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

### 2. Load Dataset

```
df = pd.read_csv("ecommerce_data.csv")
```

### 3. Convert Date Column

```
df['order_date'] = pd.to_datetime(df['order_date'])
reference_date = df['order_date'].max() + pd.Timedelta(days=1)
```

### 4. Create RFM Table

```
rfm = df.groupby('customer_id').agg({
    'order_date': lambda x: (reference_date - x.max()).days,
    'order_id': 'nunique',
    'total_sales': 'sum'
})

rfm.columns = ['Recency', 'Frequency', 'Monetary']
```

### 5. Create RFM Scores

```
rfm['R_score'] = pd.qcut(rfm['Recency'], 4, labels=[4,3,2,1])
rfm['F_score'] = pd.qcut(rfm['Frequency'].rank(method='first'), 4, labels=[1,2,3,4])
rfm['M_score'] = pd.qcut(rfm['Monetary'], 4, labels=[1,2,3,4])
```

### 6. Combine Scores & Create Segments

```
rfm['RFM_Score'] = (
    rfm['R_score'].astype(str) +
    rfm['F_score'].astype(str) +
    rfm['M_score'].astype(str)
)

def segment(row):
```

```
if row['R_score'] == '4' and row['F_score'] == '4':  
    return 'Champions'  
elif row['F_score'] == '4':  
    return 'Loyal Customers'  
elif row['R_score'] == '1':  
    return 'At Risk'  
else:  
    return 'Others'
```

```
rfm['Segment'] = rfm.apply(segment, axis=1)
```

## 7. Export Results

```
rfm.to_csv("rfm_segments.csv")
```

**Deliverables:** task15\_rfm.ipynb, rfm\_segments.csv, segment\_actions.txt