Step 1: Data Loading & Preprocessing

Dataset info:

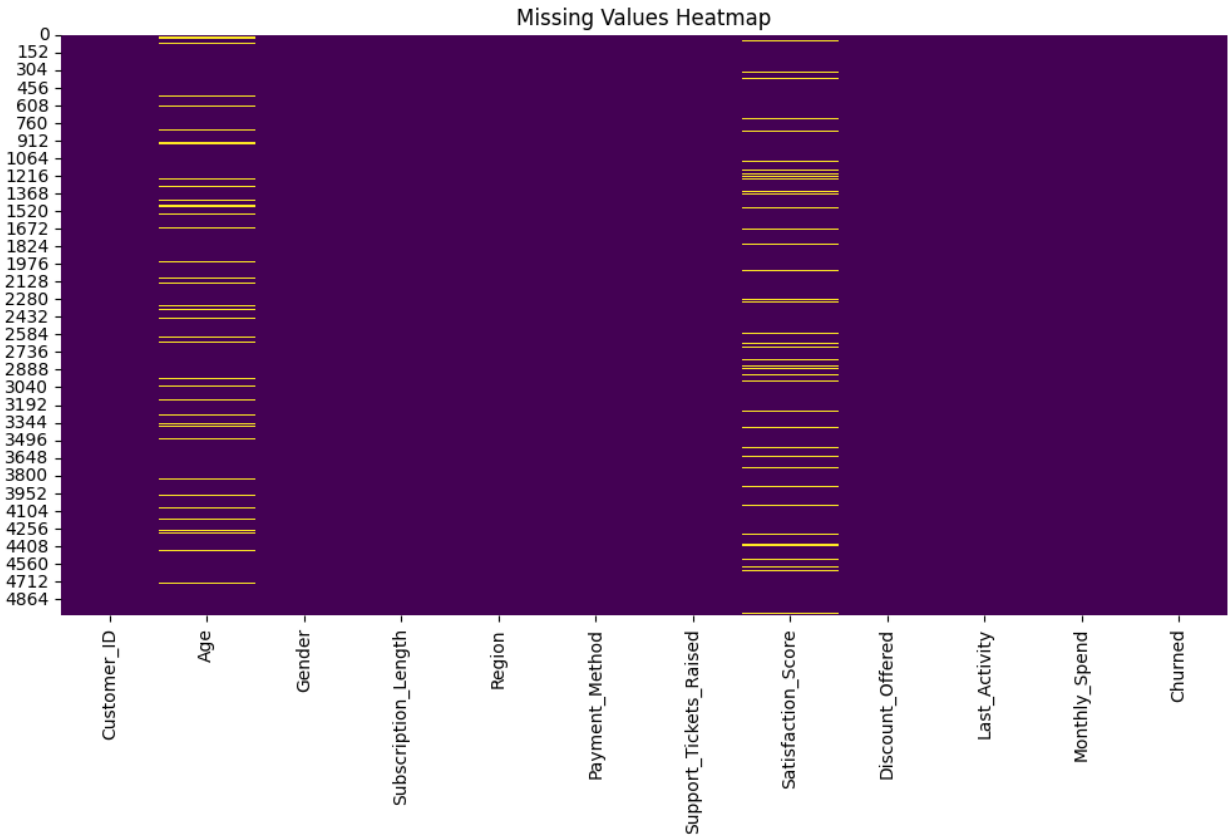
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Customer_ID             5000 non-null   object
 1   Age                     4500 non-null   float64
 2   Gender                  5000 non-null   object
 3   Subscription_Length     5000 non-null   int64
 4   Region                  5000 non-null   object
 5   Payment_Method          5000 non-null   object
 6   Support_Tickets_Raised  5000 non-null   int64
 7   Satisfaction_Score      4500 non-null   float64
 8   Discount_Offered        5000 non-null   float64
 9   Last_Activity           5000 non-null   int64
 10  Monthly_Spend           5000 non-null   float64
 11  Churned                 5000 non-null   int64
dtypes: float64(4), int64(4), object(4)
memory usage: 468.9+ KB
```

Data before cleaning:

```
Initial shape of the dataset: (5000, 12)
```

	Customer_ID	Age	Gender	Subscription_Length	Region	Payment_Method	Support_Tickets_Raised	Satisfaction_Score	Discount_Offered	Last_Activity	Monthly_Spend	Churned
0	CUST000001	56.0	Male	54	South	PayPal	0	9.0	6.42	319	62.11	1
1	CUST000002	69.0	Female	21	East	Debit Card	1	2.0	13.77	166	37.27	1
2	CUST000003	46.0	Female	49	East	PayPal	3	8.0	19.91	207	61.82	0
3	CUST000004	32.0	Male	47	West	Debit Card	3	1.0	13.39	108	40.96	1
4	CUST000005	60.0	Male	6	East	Credit Card	2	NaN	13.18	65	45.97	0

```
Missing values per column:
Age                   500
Satisfaction_Score    500
dtype: int64
```



This heatmap reveals that dataset has missing values specifically in the Age, and Satisfaction_Score columns. The other columns appear to be complete. This information is crucial before performing further data analysis or building model. However, for this project we have decided to keep the missing values because a row with a missing value in one column (like Age or Satisfaction_Score in your heatmap) still contains complete and potentially crucial information in many other columns (Subscription_Length, Region, Monthly_Spend, Churned, etc.). Deleting the entire row means discarding all that valid data, which could reduce the size of your dataset and weaken the insights or predictive power you can derive from the remaining complete columns.

Next step is to remove duplicates from dataset.

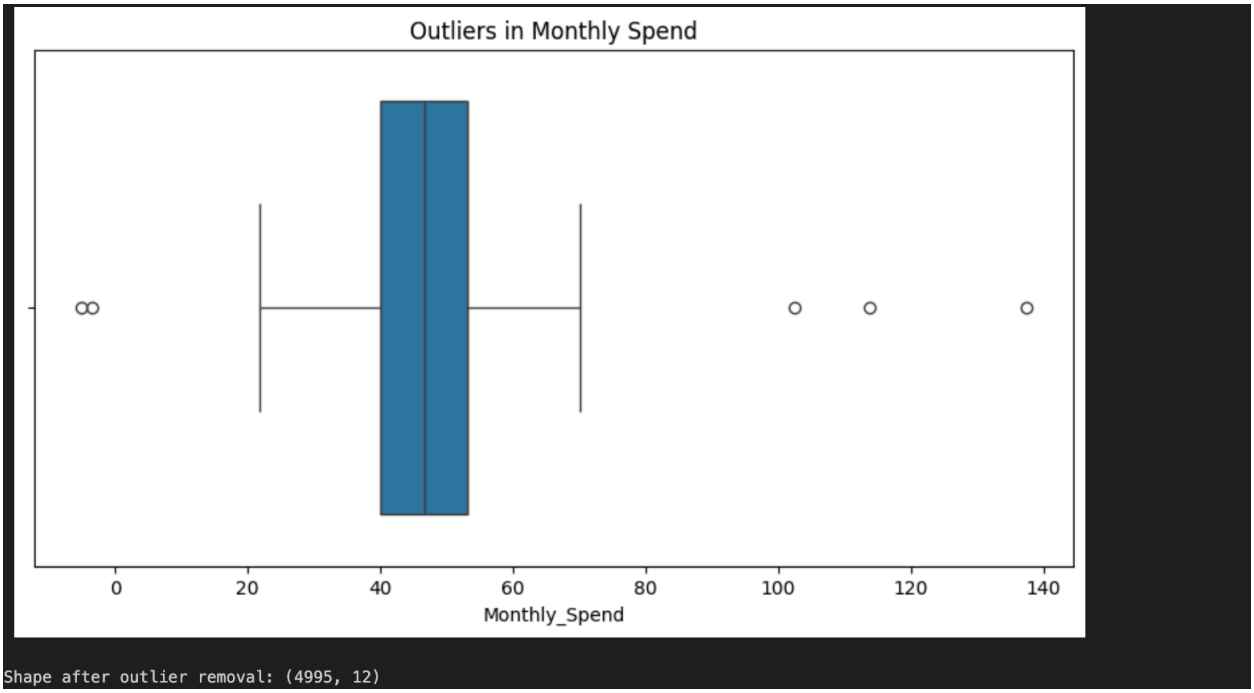
After removing duplicates

```
# ====================
# 5. Remove Duplicates
# ====================
print("Duplicate rows:", df.duplicated().sum())
df = df.drop_duplicates()
print("Shape after removing duplicates:", df.shape)
```

[5] Python

```
Duplicate rows: 0
Shape after removing duplicates: (5000, 12)
```

Outliers detection and removal



The box plot shows the distribution of 'Monthly_Spend' and identifies outliers. The individual points plotted to the right of the right whisker are the outliers in the 'Monthly_Spend' data. These points represent values that are significantly higher than the rest of the data.

Outliers were removed from the dataset. This decision was made to improve the performance and reliability of models aimed at predicting subscriptions users are less likely to use. Since low subscription usage is expected to correlate with lower monthly spending, the presence of extreme high spending values can disproportionately skew model training, obscuring the subtle patterns and characteristics associated with low-usage behaviors. Removing these high outliers allows the predictive model to better focus on and learn from the distribution of typical and low spending data points, thereby enhancing its ability to accurately identify users less likely to utilize their subscriptions.

Saved cleaned data to data folder under name cleaned_data.csv

Step 2: Data Analysis & Visualization

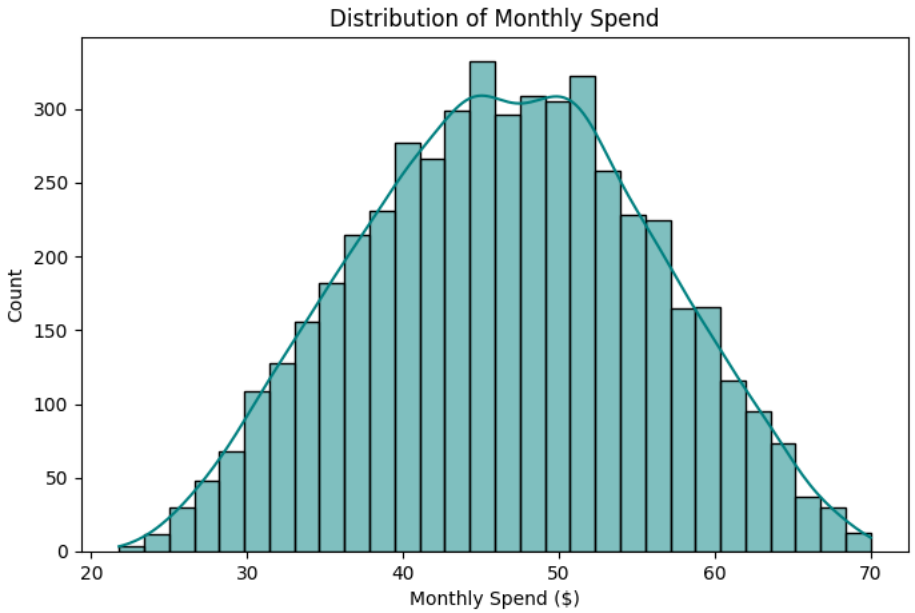
Descriptive Stats:

Average Monthly Spend: 46.59741541541542

Average Subscription Length: 29.71091091091091

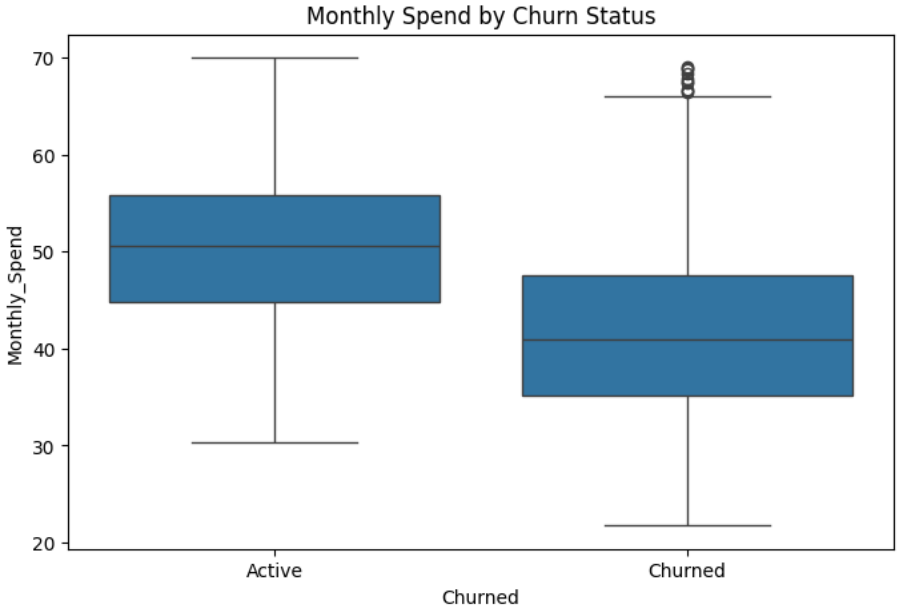
Average Satisfaction Score: 5.546607341490545

Spending Patterns:



Histogram of Monthly Spend: The histogram of Monthly Spend appears to be approximately normally distributed. It is roughly symmetrical with a peak around $50. There is no significant skew to the left or right. The distribution resembles a bell curve, which is characteristic of a normal distribution.

Spend by Gender: The box plot comparing monthly spend between males and females shows remarkable similarity. Both genders exhibit very close median spending values, similar interquartile ranges (indicating comparable spread in the middle 50% of spenders), and similar overall ranges of spending as shown by the whiskers.



Spend comparison between churned and active users: The analysis reveals a distinct difference in monthly spending based on churn status. Customers who churn tend to have lower monthly spending compared to customers who remain active. This suggests that lower monthly spending is associated with a higher propensity to churn, although there are some exceptions as indicated

by the high-spending outliers in the churned group. This finding indicates that monthly spend is a potential factor related to customer churn

T-test result

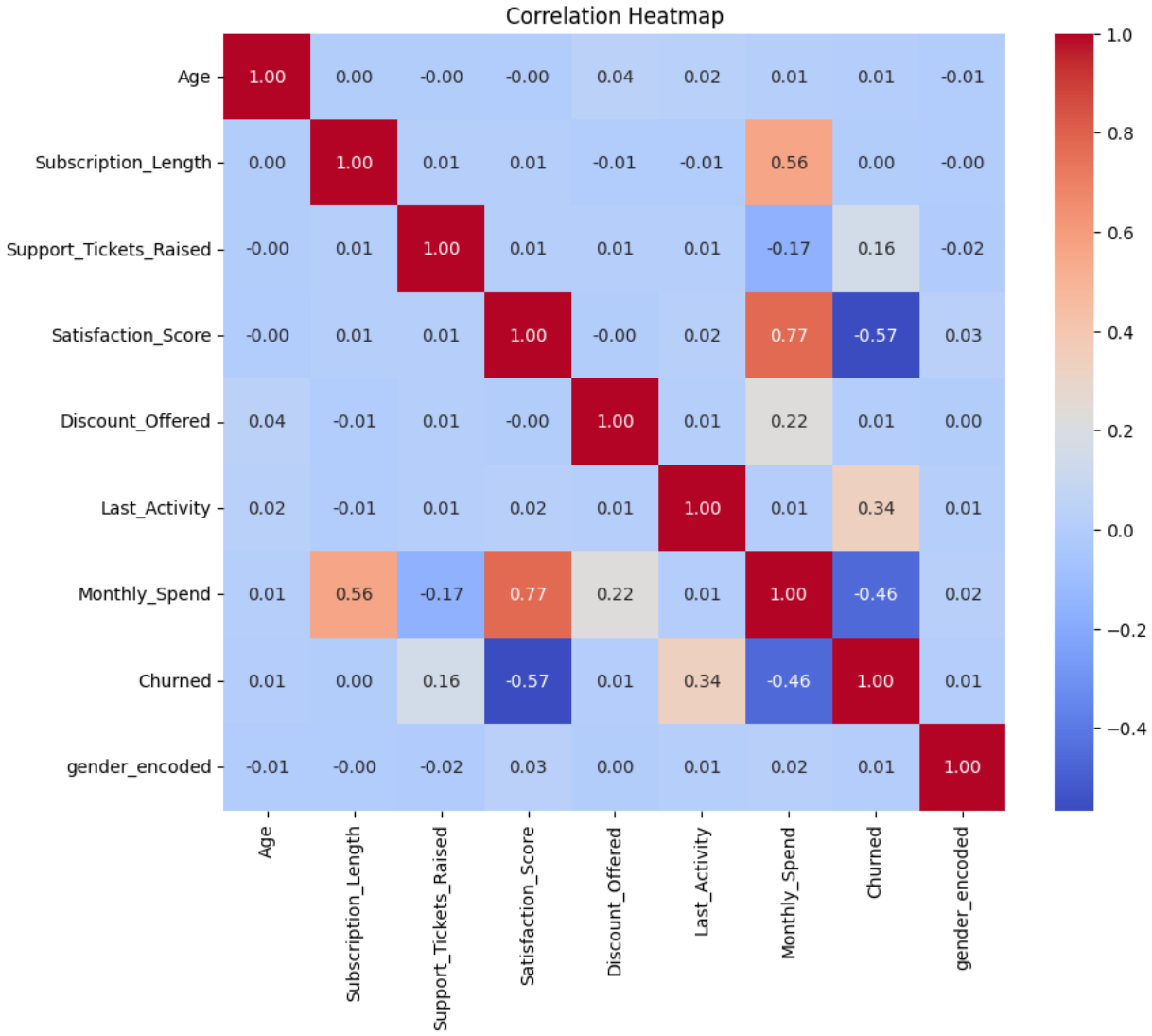
T-test: t = -35.730, p = 0.0000

Since the p-value is **less than 0.05**, we **reject the null hypothesis**.
This means there **is a statistically significant difference** in monthly spending between churned and non-churned users.

In fact, the large negative t-value indicates that **churned users tend to spend significantly less** than active users. This supports the idea that **low spenders are more likely to churn**, which can guide predictive modeling and recommendation strategies later in the project.

Correlation Analysis

- Correlation heatmap



Key Insights from the Heatmap:

- **Churn vs. Satisfaction Score**:
  - **Correlation = -0.57**
  - Interpretation: As satisfaction **decreases**, likelihood of churn **increases** significantly.
  - This is the **strongest negative correlation** with churn crucial for prediction.
- **Churn vs. Monthly Spend**:
  - **Correlation = -0.46**

    - Interpretation: Lower monthly spend is **associated with higher churn**. Supports findings from t-test.

- **Monthly Spend vs. Satisfaction Score**:
    - **Correlation = 0.77**
    - Interpretation: Happier users tend to **spend more**. Strongest **positive correlation**.
- **Monthly Spend vs. Subscription Length**:
    - **Correlation = 0.56**
    - Interpretation: Long-term subscribers tend to **spend more** each month.
- Most other features (Age, Gender, Support Tickets, etc.) show **weak or no correlation** with churn or spending.

Step 3: Machine Learning & Prediction

This step focused on using machine learning techniques to predict customer behavior in relation to subscription services. The two main tasks were:

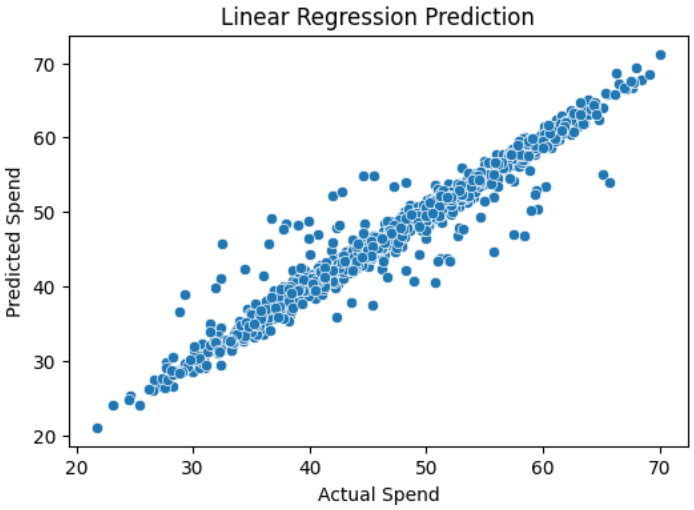
1. **Churn Prediction (Classification)** – Identify whether a customer is likely to cancel their subscription.
2. **Monthly Spend Prediction (Regression)** – Forecast how much a customer is expected to spend on subscriptions.

Part 1: Monthly Spend Prediction (Linear Regression)

To predict how much a customer is likely to spend monthly on subscriptions, we used a **Linear Regression** model.

Model Performance:

- **Root Mean Squared Error (RMSE):** *12.45*



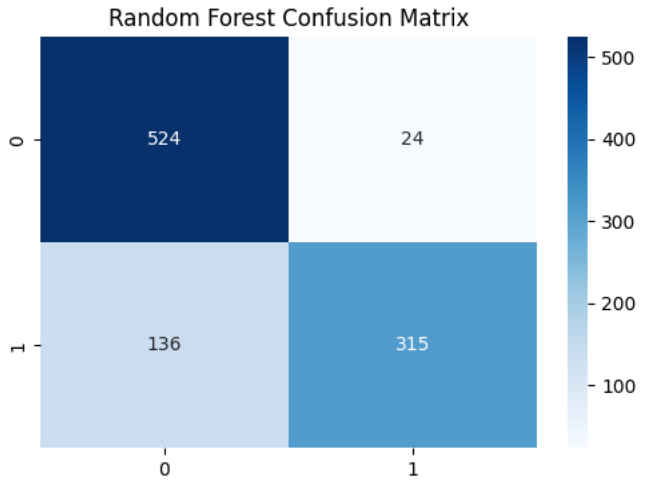
Part 2: Churn Prediction (Logistic Regression)

To predict whether a customer is likely to **cancel their subscription**, we used a **Logistic Regression**classification model.

Model Performance:

- **Accuracy:** 83.25%
- **Precision (Class 1 - Churned):** 92.9%
- **Recall (Class 1 - Churned):** 69.9%
- **F1-score (Class 1):** 79.5%

Confusion Matrix:



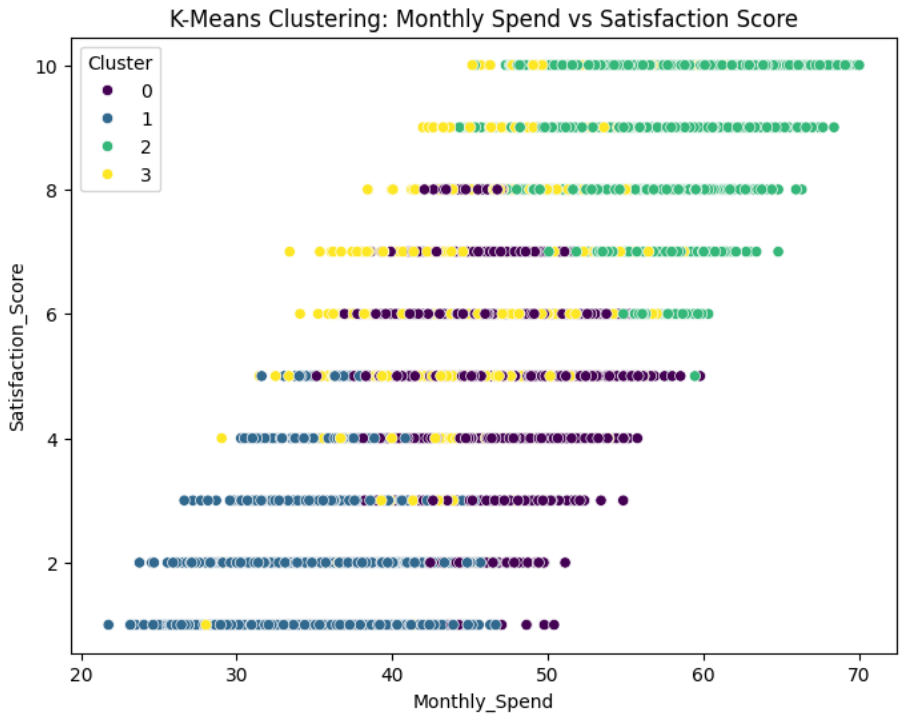
The model does a solid job detecting churn, with a high **precision**, meaning when it predicts churn, it's often correct. Some churners are missed (as shown by false negatives), but overall the balance between precision and recall is strong.

Step 4:

Exploratory Data Insights: Customer Segmentation & Behavior Patterns

This section explores key customer patterns using **K-Means Clustering** and **Association Rule Mining**, focusing on identifying distinct user segments and discovering relationships between categorical customer attributes. These insights support actionable decision-making in areas such as customer targeting, product recommendations, and retention strategies.

1. Customer Segmentation via K-Means Clustering



A **K-Means clustering algorithm** was applied to the dataset using two continuous variables: Monthly Spend and Satisfaction Score. The goal was to discover natural groupings in customer behavior based on spending and satisfaction.

Cluster Plot Overview:

- The scatter plot uses:
    - **X-axis:** Monthly Spend

- **Y-axis:** Satisfaction Score
- **Color-coded points:** Each representing one of the 4 clusters discovered

Interpretation of Clusters:

- **Cluster 0:** A mix of spenders with satisfaction scores spread across low to moderate values.
- **Cluster 1:** High spenders with consistently high satisfaction **ideal loyal customers**.
- **Cluster 2:** Low to moderate spenders with low satisfaction **potential churn risks**.
- **Cluster 3:** Moderate spenders with high satisfaction possibly **value-seeking loyalists**.

Takeaway:

This clustering helps define customer **personas**. For instance:

- **Cluster 1** customers might be ideal for **VIP programs or upselling**.
- **Cluster 2** should be the focus of **retention and satisfaction improvement efforts**.
- Such segmentation provides a foundation for **personalized marketing strategies**.

Model Training:

1. Logistic Regression for Churn Prediction

- Predict whether a customer is likely to churn (cancel their subscription).
- **Input**:
  - Monthly_Spend
  - Support_Tickets_Raised
  - Satisfaction_Score
  - Discount_Offered
  - Subscription_Length
- **Target**: Churned (0 = No, 1 = Yes)
- Provided a simple but interpretable model to give real-time churn predictions.

2. Linear Regression for Monthly Spend Estimation

- Predict the expected monthly spend based on user behavior and satisfaction.

- **Input Features**:
    - Subscription_Length
    - Support_Tickets_Raised
    - Satisfaction_Score
    - Discount_Offered
    - Encoded Payment_Method
    - Encoded Region
- **Target**: Monthly_Spend
- This model estimates what a similar customer would typically spend, useful for comparison or upselling analysis.

3. KMeans Clustering for Customer Segmentation

- Segment customers into distinct groups based on behavior and satisfaction.
- **Input Features** (scaled):
    - Same features as spend model
- **Output**: Cluster labels (0, 1, 2, ...)
- Helped define customer types like “High Satisfaction, Low Spend” or “At-Risk, High Discount Reliant.”

Model Training Techniques

- All models were trained using scikit-learn.
- **Preprocessing** included:
    - Label encoding for Gender, Payment_Method, and Region.
    - Standardization using StandardScaler.
- Models were evaluated using accuracy (for churn), mean squared error (for spend), and silhouette score (for clustering).

Goals of models:

- Provided **individual churn insights** to help users rethink their subscriptions.

- Gave **estimated spend values** to help users evaluate whether they're spending more than similar users.
- Segmented customers for a **personalized understanding** of where they stand among peer groups.